



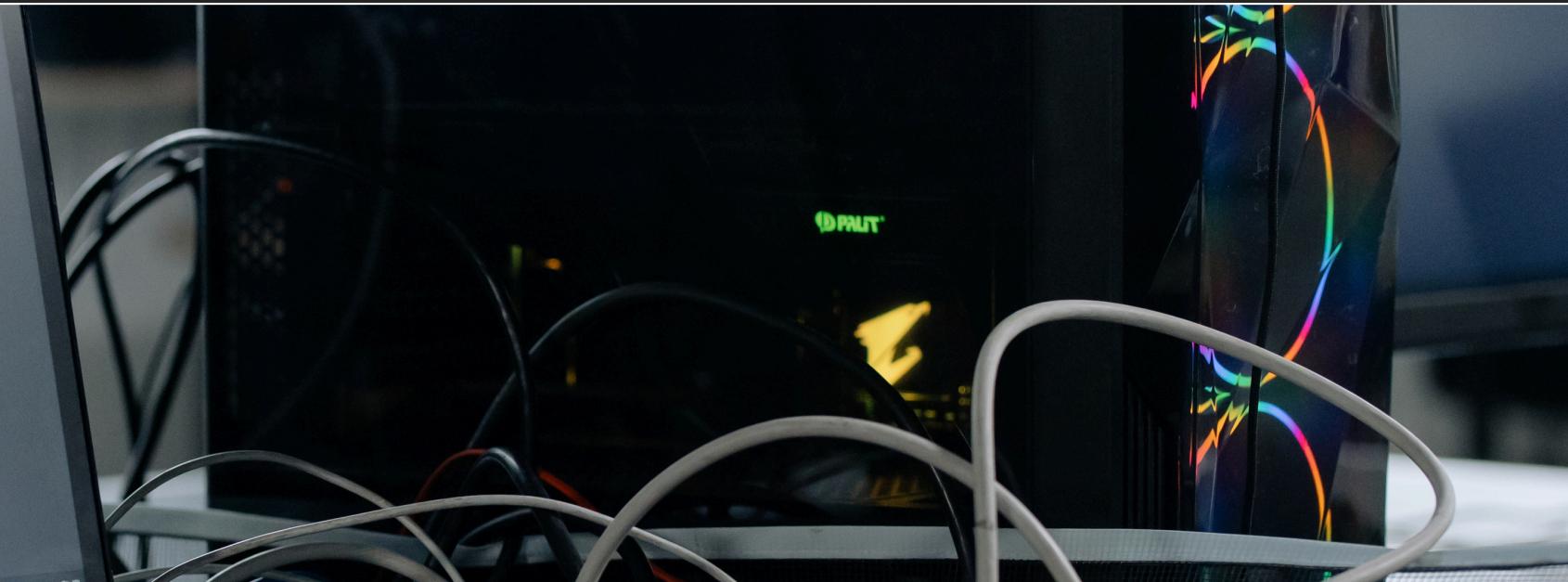
Search



Basic C-lang Camp

Your Journey to Becoming a Inspire Programmer

BEGINNER TO ADVANCED



Let's get to know each other first



P' BLUE



P' POND

BACHELOR COMPUTER ENGINEERING

King Mongkut's Institute Of Technology Ladkrabang
(KMITL)



02

Types of Programmers

Explore Different Roles

Front-end Developer

A web or application development specialist who can directly see and interact with the user in order to control the operation of the system. Such as HTML, CSS, and Javascript.

Back-end Developer

Work with the backend (e.g. databases, servers, business logic) Create a stable, secure, and efficient system for data management.

Full-Stack Developer

Can work on both Frontend and Backend. This makes being a full-stack developer a must-have skill set, including social skills sometimes also need to be able to communicate and coordinate with clients, users, or team

Mobile Developer

Develop systems for mobile devices such as smartphones and tablets (both iOS and Android). The main goal is to create applications that are easy to use and efficient on mobile platforms.



03

What is a C Language?

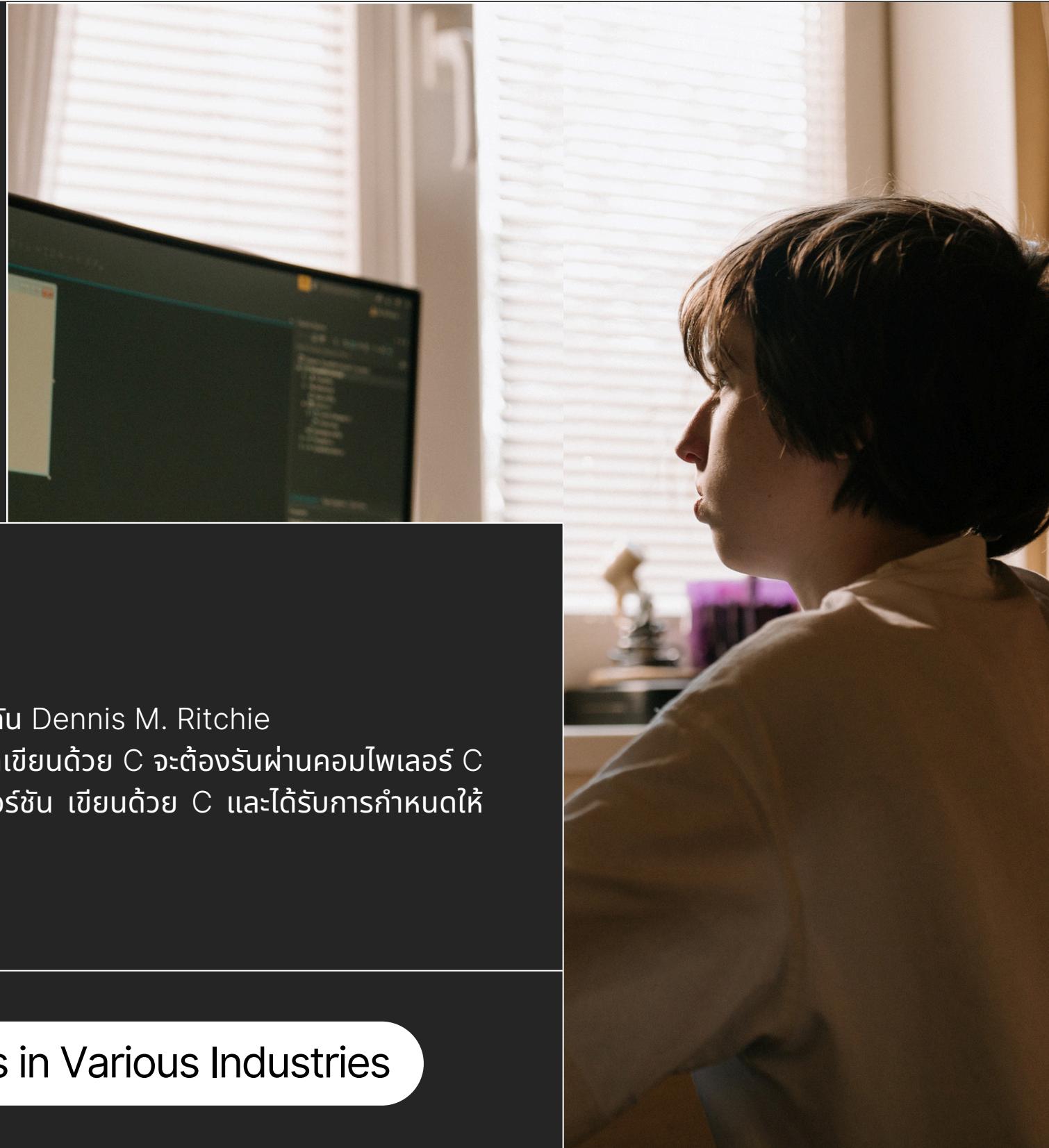
Understanding the Origin

ภาษาโปรแกรม C เป็นที่พัฒนาขึ้นในช่วงต้นทศวรรษ 1970 โดยนักวิทยาศาสตร์คอมพิวเตอร์ชาวอเมริกัน Dennis M. Ritchie เป็นภาษาเชิงกระบวนการและวัตถุประสัมภ์ทั่วไปที่ให้การเข้าถึงหน่วยความจำระบบในระดับต่ำ โปรแกรมที่เขียนด้วย C จะต้องรันผ่านคอมพิวเตอร์ C เพื่อแปลงเป็นไฟล์ปฏิบัติการที่คอมพิวเตอร์สามารถรันได้ ระบบปฏิบัติการ (OS) ที่ใช้ Unix หลายเวอร์ชัน เขียนด้วย C และได้รับการกำหนดให้เป็นส่วนหนึ่งของ Portable Operating System Interface (POSIX)

Develops Software

Builds Applications

Works in Various Industries

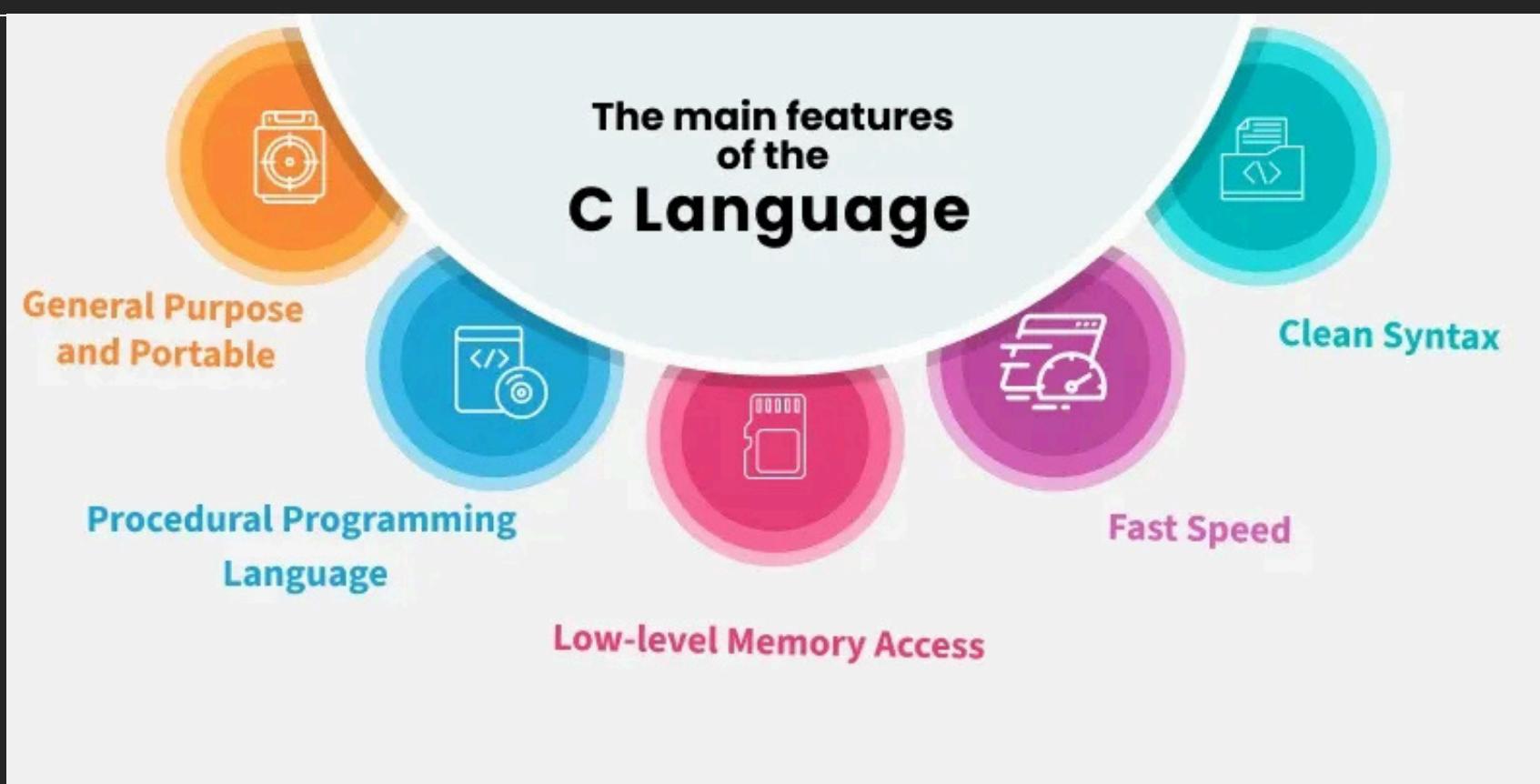


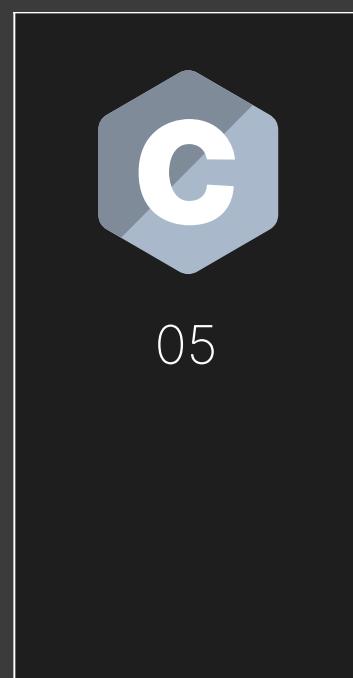


04

Key Skills for C Language

Master the Essentials





C compiler

compiler C

ทั้งหมด รูปภาพ วีดีโอ ข้อมูล วีดีโอลิสต์ ข่าวสาร หนังสือ เพิ่มเติม ▾

 [Programiz](https://www.programiz.com)
https://www.programiz.com › online-co... · แปลงหน้า ·

Online C Compiler
Write and run your C programming code using our online compiler. Enjoy additional features like code sharing, dark mode, and support for multiple languages.

[C++ Compiler](#) [Python Compiler](#) [Online Java Compiler](#)

 [GDB online Debugger](https://www.onlinegdb.com)
https://www.onlinegdb.com › online_c... · แปลงหน้า ·

Online C Compiler - online editor
Code, Compile, Run and Debug C program online. Write your code in this editor and press "Run" button to compile and execute it. * ...

 [OneCompiler](https://onecompiler.com)
https://onecompiler.com › c · แปลงหน้า ·

C Online Compiler
Write, Run & Share C Language code online using OneCompiler's C online compiler for free. It's one of the robust, feature-rich online compilers for C language

 [W3Schools](https://www.w3schools.com)
https://www.w3schools.com › c › c_co... · แปลงหน้า ·

C Online Compiler (Editor / Interpreter)
C Compiler (Editor) With our online C compiler, you can edit C code, and view the result in your browser.

โปรแกรมที่ช่วยให้คุณสามารถเขียน โปรแกรมภาษา C แก้ไข และรันโปรแกรม
เหล่านั้นได้โดยตรงบนเว็บเบราว์เซอร์ โดยไม่ต้องติดตั้งคอมพิวเตอร์ C หรือ
สภาพแวดล้อมการพัฒนาแบบรวม (IDE) ลงบนเครื่องคอมพิวเตอร์ของคุณ



www.onlinegdb.com/online_c_compiler

OnlineGDB
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login

```
main.c
1 // *****
2 // Online C Compiler.
3 // Code, Compile, Run and Debug C program online.
4 // Write your code in this editor and press "Run" button to compile and
5 // *****
6
7
8
9 #include <stdio.h>
10
11 int main()
12 {
13     printf("Hello World");
14
15     return 0;
16 }
```

C Compiler

◀ Previous Next ▶

C Compiler (Editor)
With our online C compiler, you can edit C code, and view the result in your browser.

#include <stdio.h>
int main() {
 printf("Hello World!");
 return 0;
}

Try it Yourself

Hello World!

Click on the "Try it Yourself" button to see how it works.

Programiz
C Online Compiler

Never struggle with DSA again.
Learn with interactive visuals.

main.c
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int main()
5 {
6 // Write C code here
7 printf("Try programiz.pro");
8
9 return 0;
}

Run Output



05

Vscode setup for C

A screenshot of a Google search results page. The search bar at the top contains the query "mingw". Below the search bar, there are tabs for All, Images, Videos, Shopping, Forums, News, Web, and More. The first result is for "mingw-w64" with a link to <https://www.mingw-w64.org>. The second result is for "MinGW - Minimalist GNU for Windows" with a link to <https://sourceforge.net/projects/mingw/>. The "MinGW" result is highlighted with a red rounded rectangle.

- ໃຊ້ web browser ດັວກ “mingw” ເລືອນ link ຕາມນີ້ <https://sourceforge.net/projects/mingw/>



05

Vscode setup for C

The screenshot shows a web browser window with the URL <https://sourceforge.net/projects/mingw/>. The page is for the MinGW project on SourceForge. The main content area displays the MinGW logo, the title "MinGW - Minimalist GNU for Windows", a brief description, and a "Download" button. The "Download" button is highlighted with a red border.

https://sourceforge.net/projects/mingw/

SOURCEFORGE

Business Software Open Source Software SourceForge Podcast Resources

Home / Open Source Software / Software Development / Build Tools / MinGW - Minimalist GNU for Windows

MinGW - Minimalist GNU for Windows

A native Windows port of the GNU Compiler Collection (GCC)
Brought to you by: [cstraus](#), [earnie](#), [gressett](#), [keithmarshall](#)

★★★★★ 206 Reviews

Downloads: 3,364,135 This Week

Download

Share This

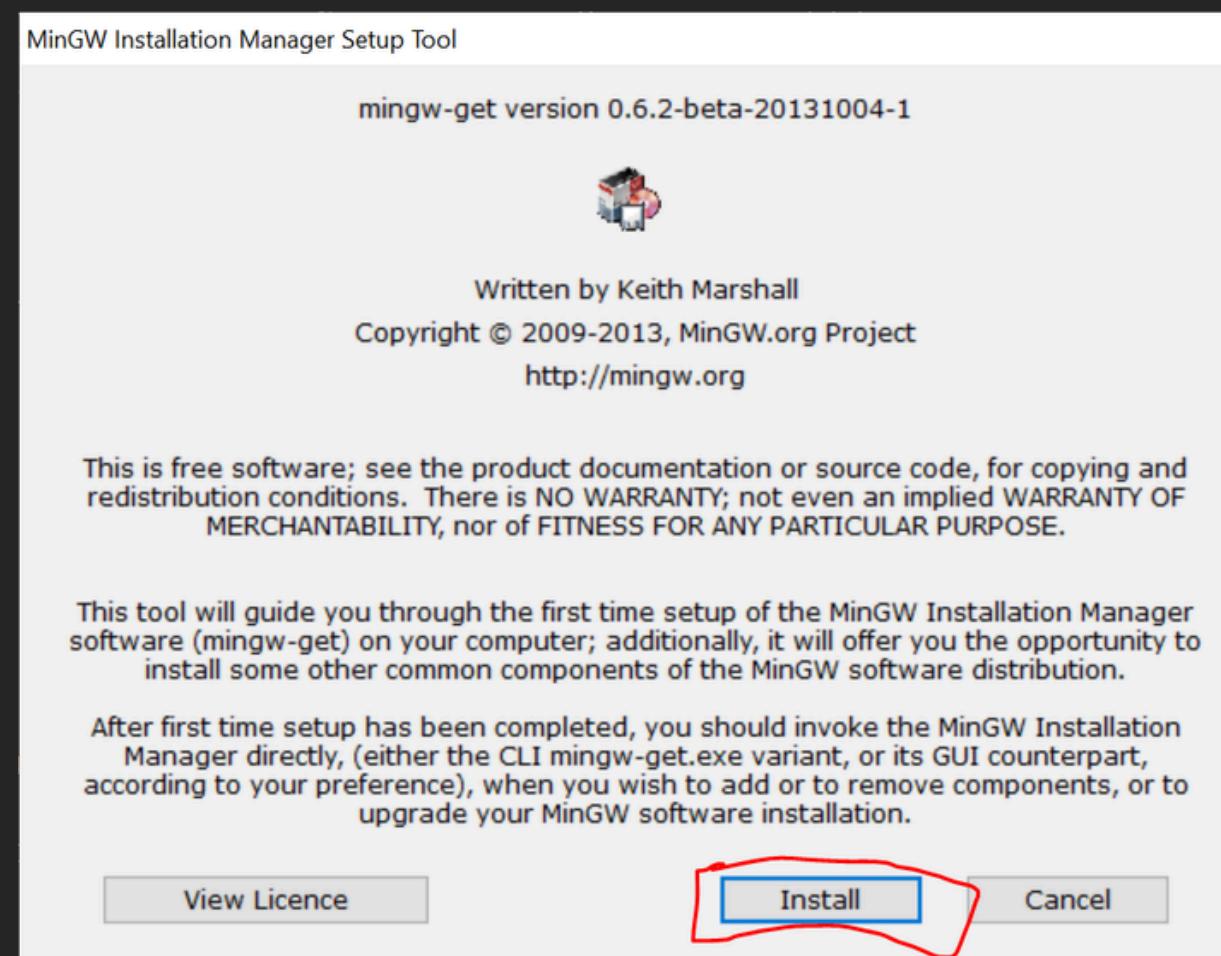
- ካልሆነ Download
- ሳይንበኩ ደንብ የሚታወቁ mingw-get-setup.exe



05

Vscode setup for C

Name	Date modified	Type	Size
▼ Today (5)			
mingw-get-setup	7/7/2025 10:43 PM	Application	85 KB

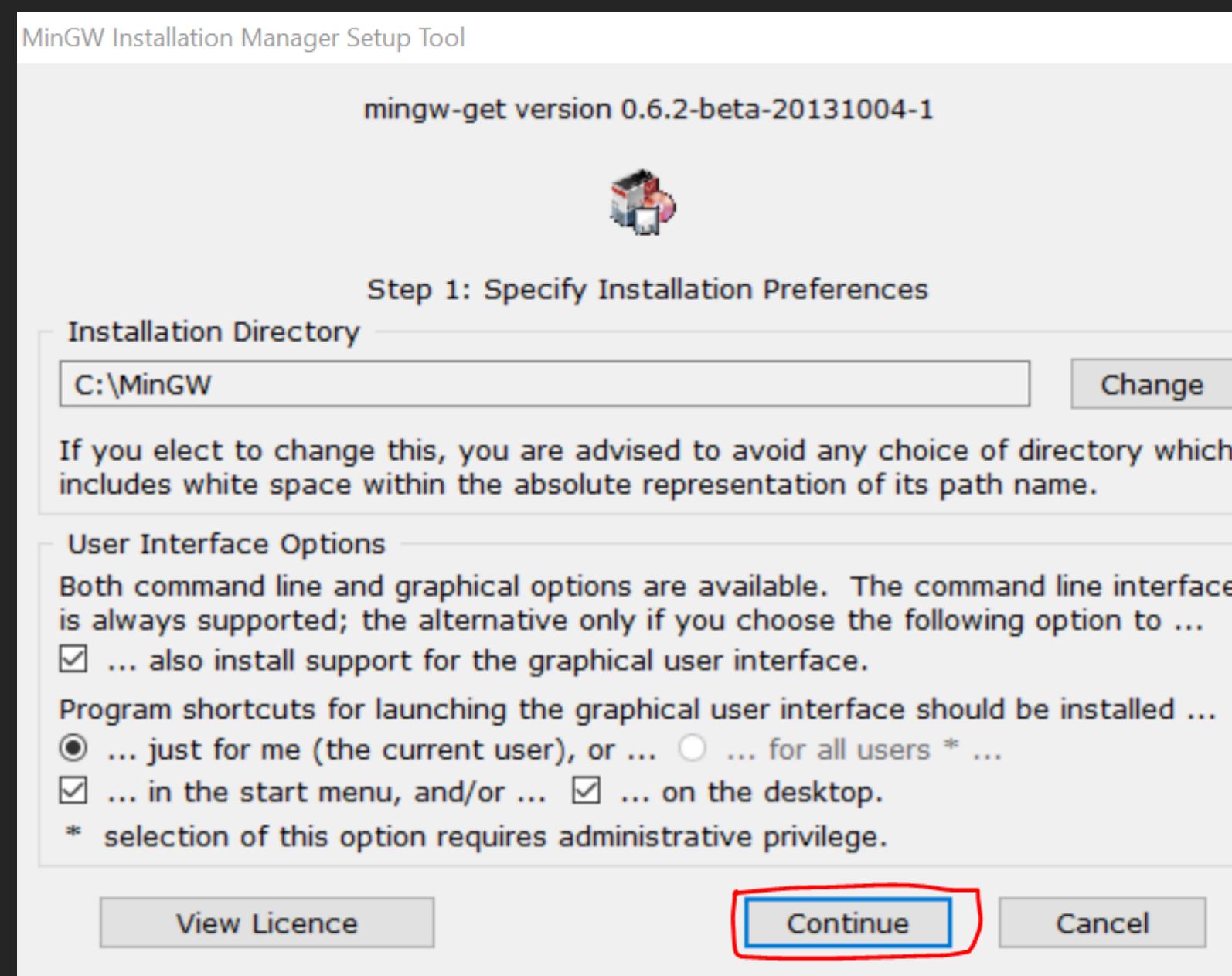


- Double Click ឲ្យមើល
- no Install



05

Vscode setup for C

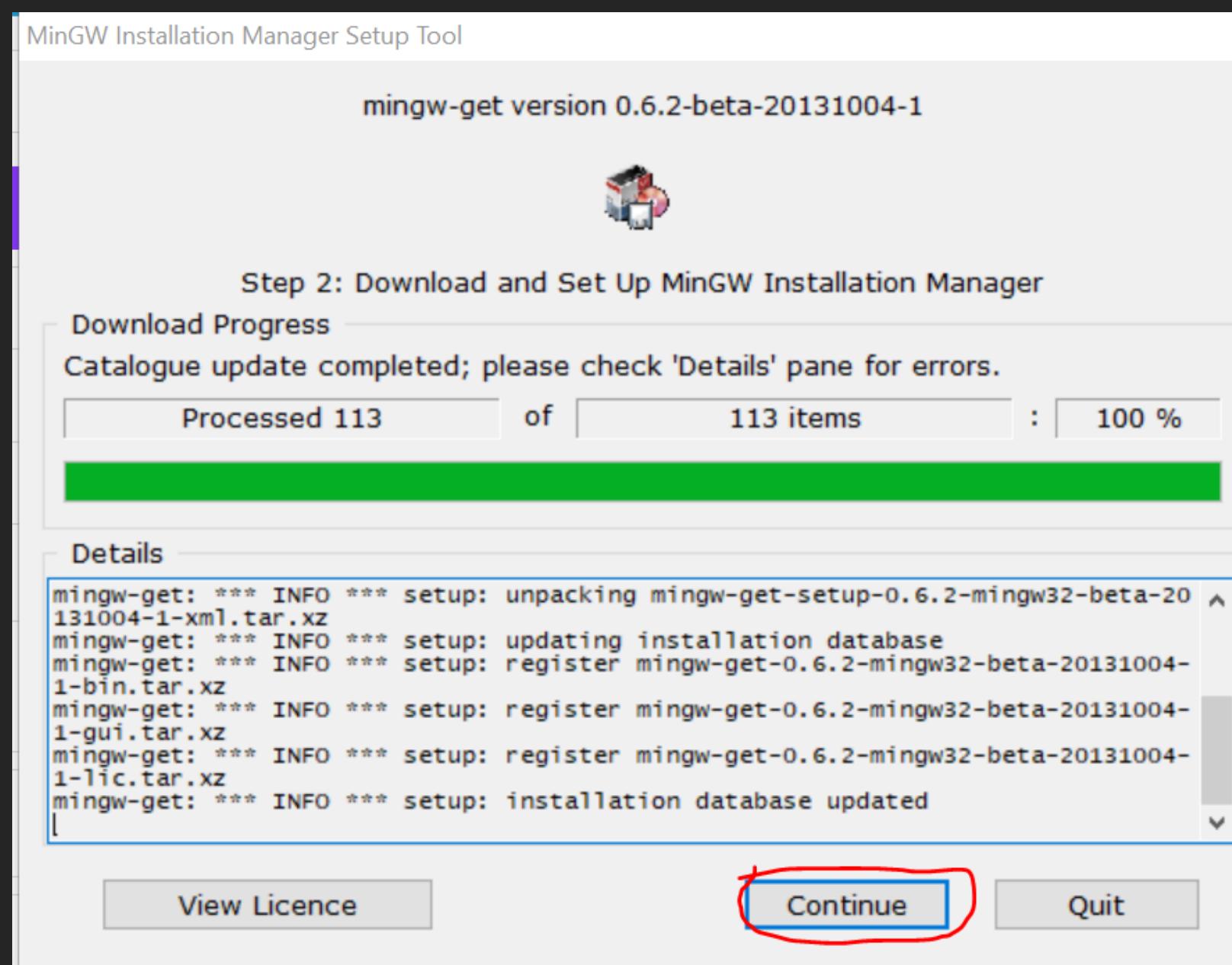


- **na Continue**



05

Vscode setup for C



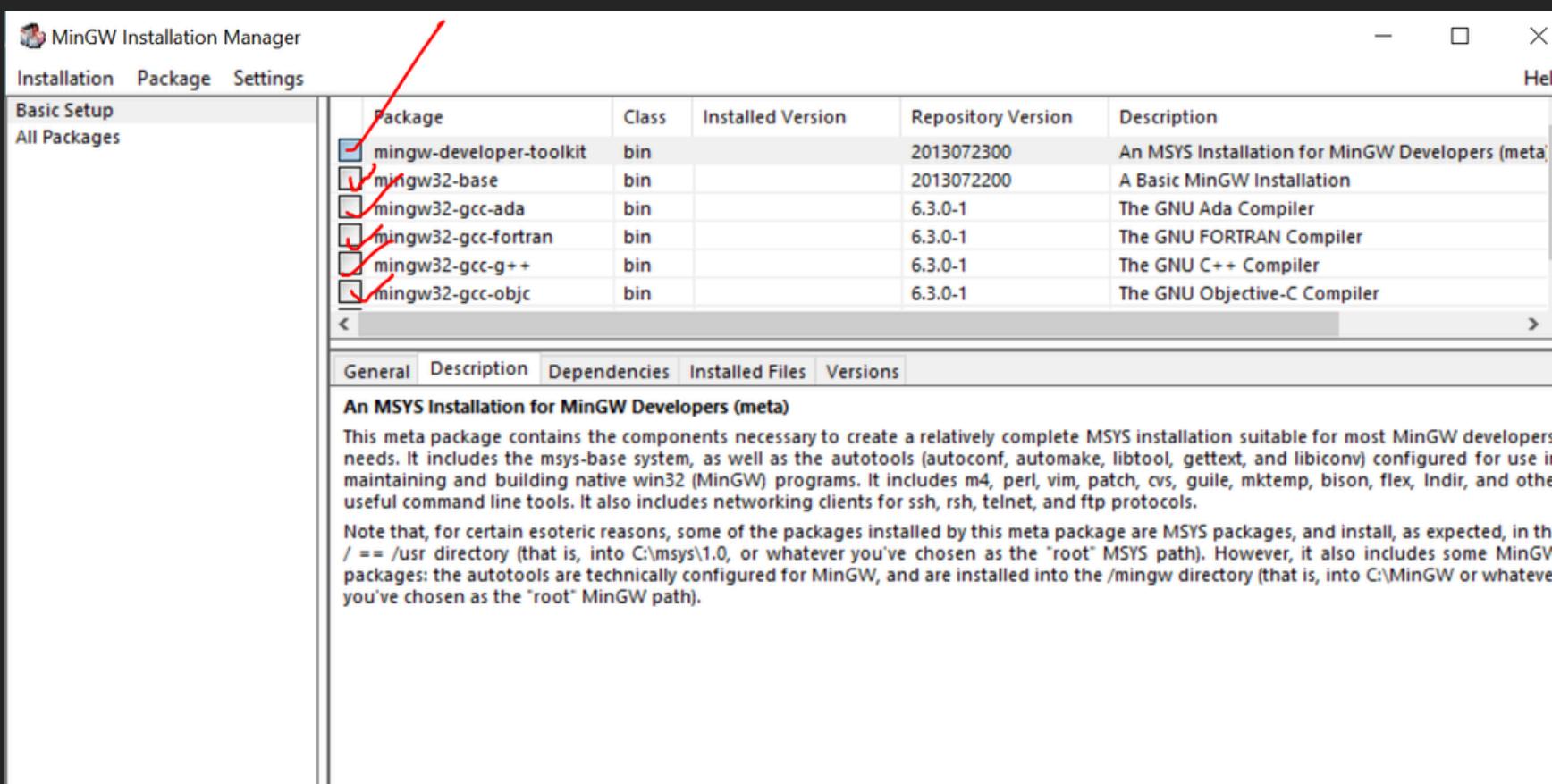
- **na Continue**



05

Vscode setup for C

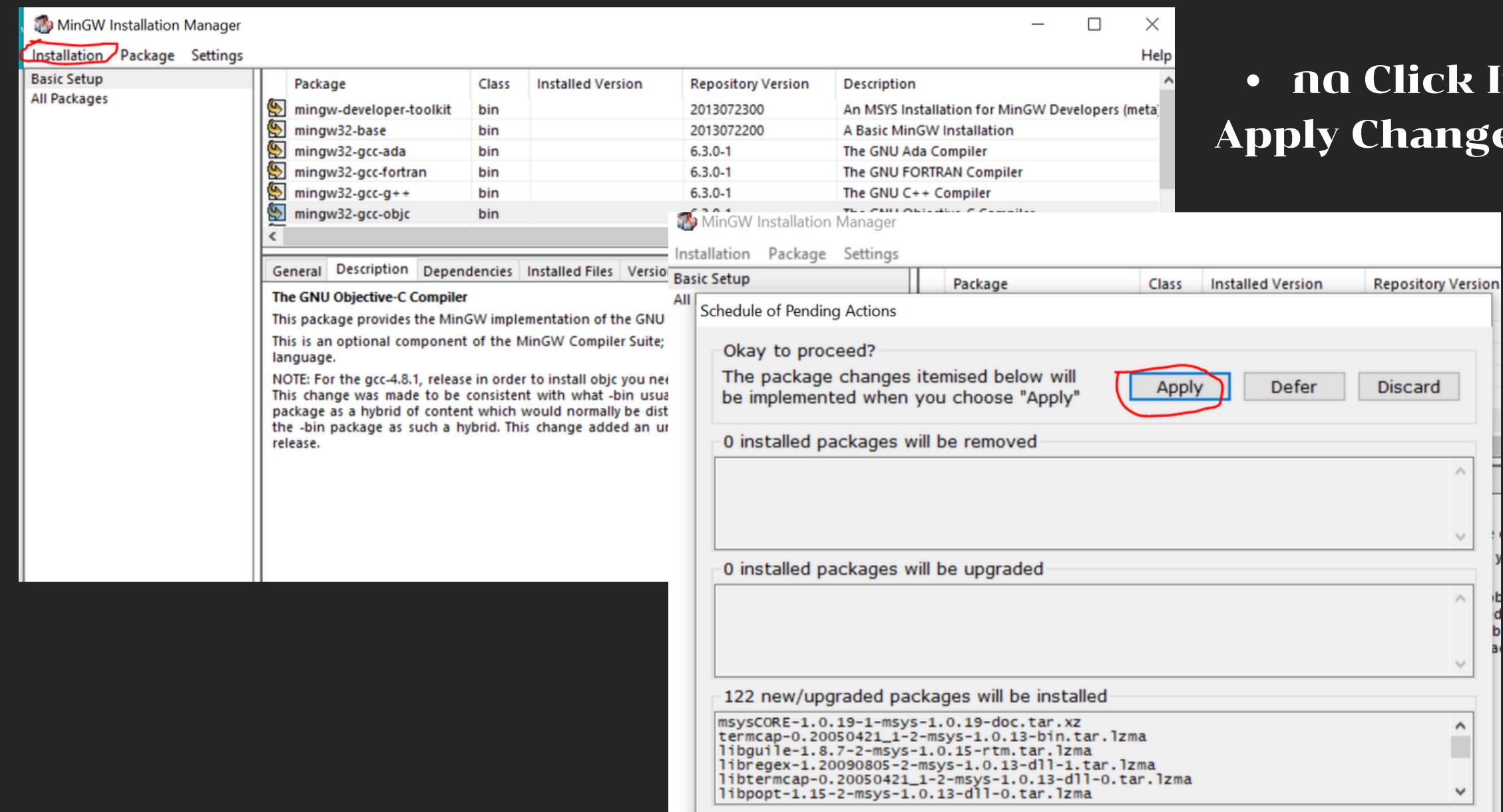
- **no Click Mark for installation នៅក្នុង**





05

Vscode setup for C

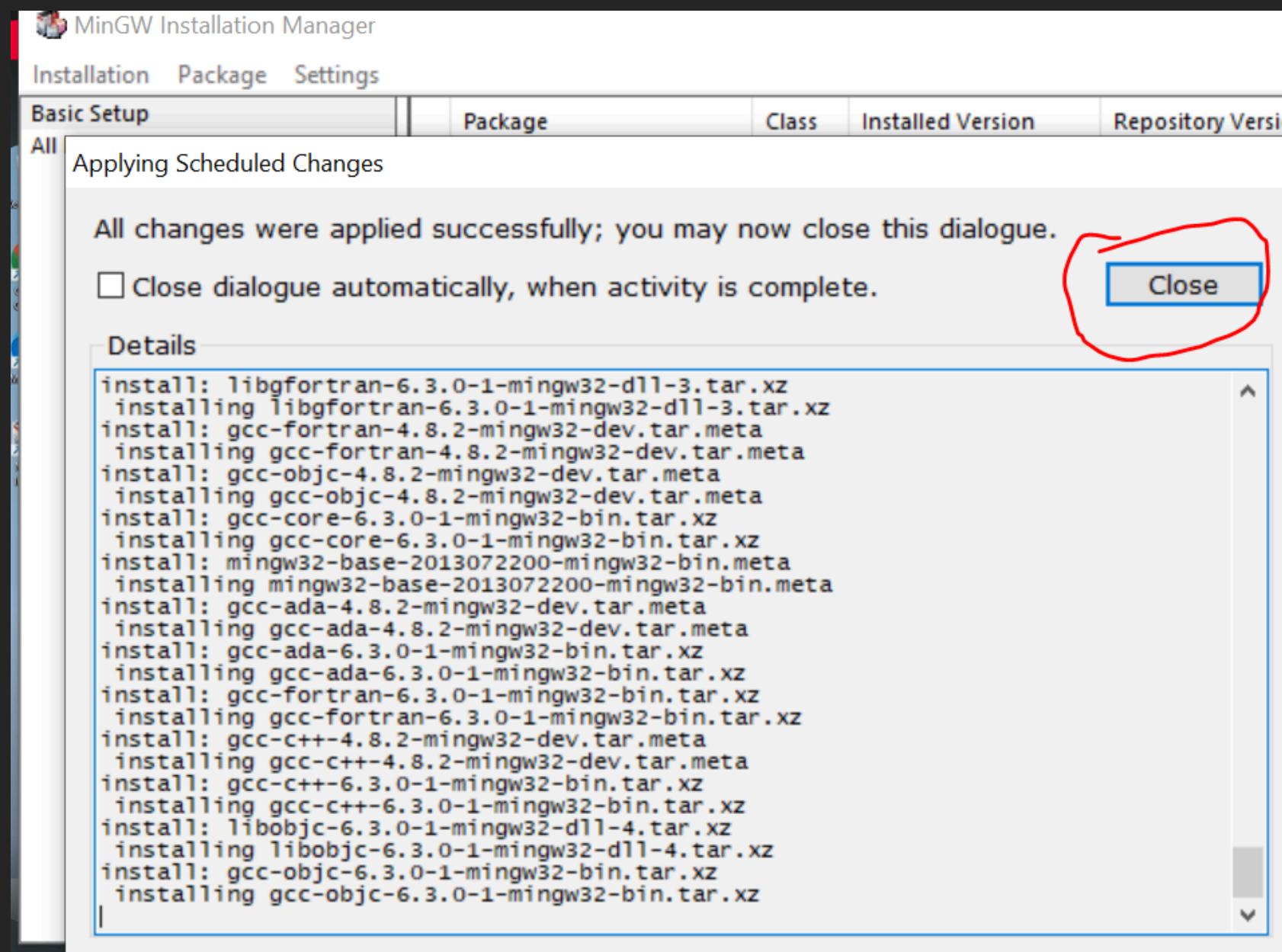


- **na Click Installation ຈານນີ້ນາ
Apply Changes ແລະປຸມ apply ອັນນີ້**



05

Vscode setup for C

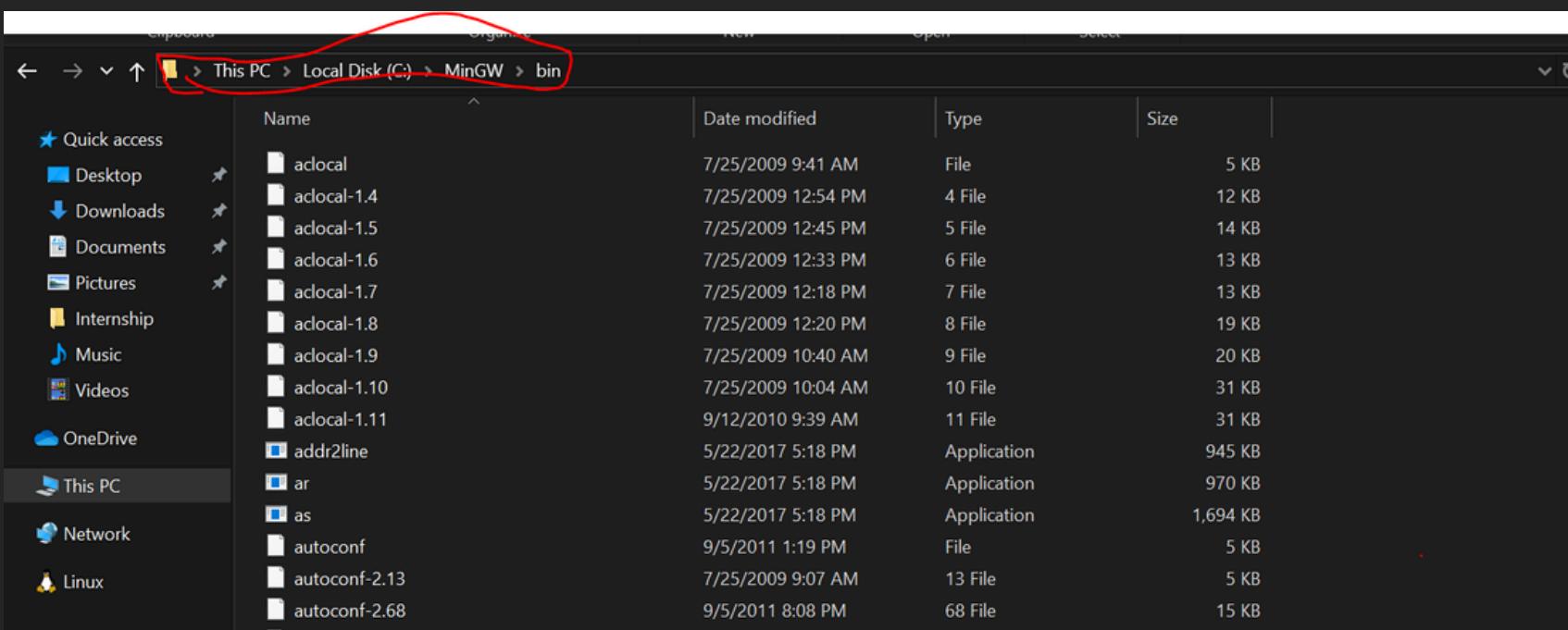


- **na Close**



05

Vscode setup for C

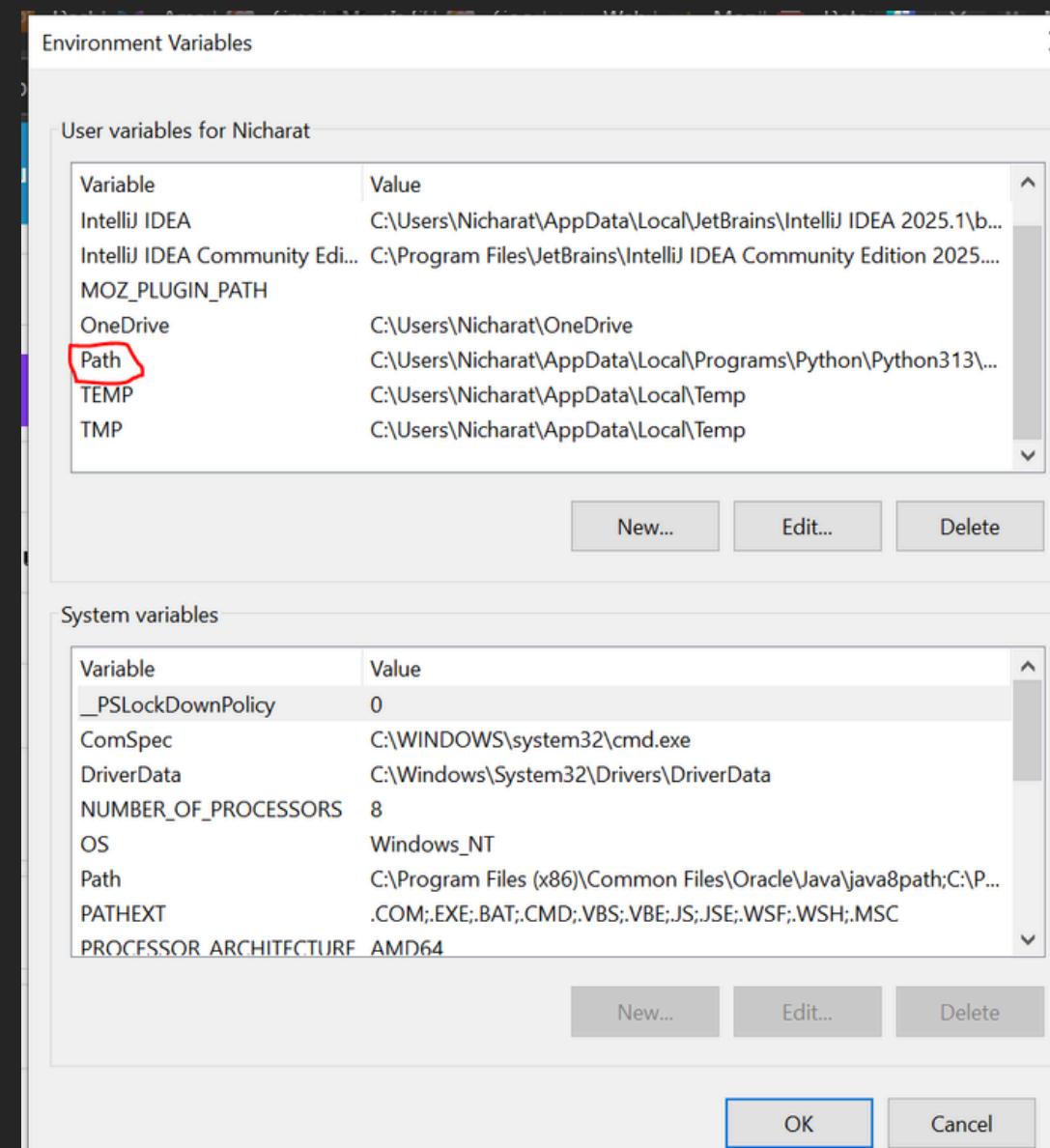


- ไปที่ file explorer → C: MinGW → bin
แล้วกดค้างที่วง งานนั้นๆ Ctrl + C (copy path)



05

Vscode setup for C



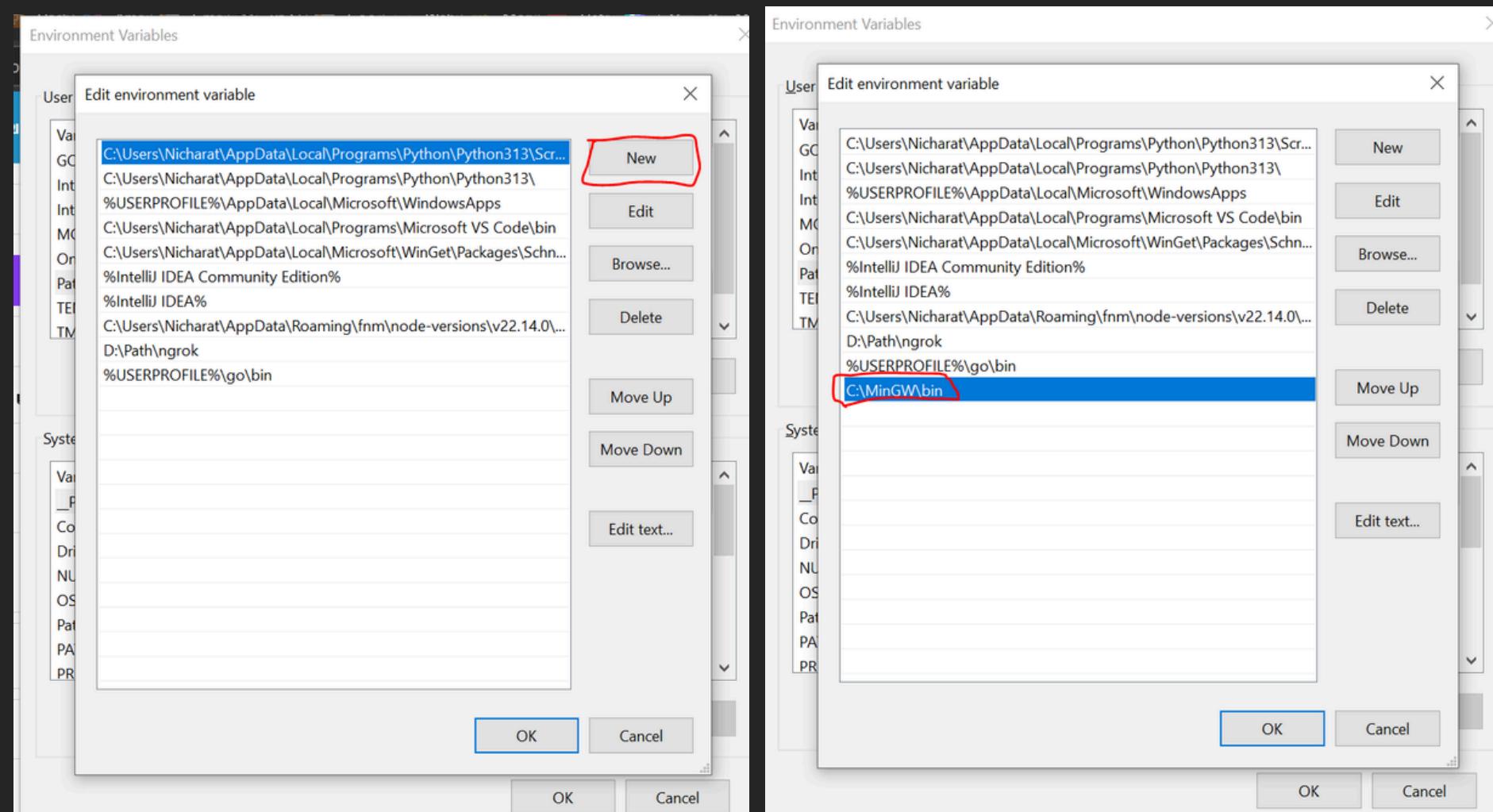
- นาไปม windows ค้นหา **Edit Environment Variables for your account** แล้วกดเลือก no click Path



05

Search

Vscode setup for C



- នានា New បាត់គ្រាង path នូវយោង copy មានការបញ្ចប់ OK ដោយមានការផ្តល់ចិត្ត



05

Vscode setup for C

```
Windows PowerShell
PS C:\Users\reddacted> gcc --version
gcc.exe (MinGW.org GCC-6.3.0-1) 6.3.0
Copyright (C) 2016 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

- ເຊັ່ນ terminal ພົມວ
gcc --version ເປັນອັນເສຣືຈສໍາຮັບ
download c complier



05

Vscode setup for C

← ⌘ https://www.google.com/search?q=visual+studio+code&sca_esv=33439f7eba77e2ac&sxsrf=AE3TifOL-tOVqrAT

Google visual studio code

All Images Videos Shopping Forums Web News More

Visual Studio Code <https://code.visualstudio.com> :

Visual Studio Code - Code Editing. Redefined

Fully customizable. Customize your **VS Code** UI and layout so that it fits your coding style. Color themes let you modify the colors in **VS Code's** user interface ...

Download Download Visual Studio Code. Free and built on open source ...

macOS Install VS Code on macOS · Download Visual Studio Code ...

Windows Use the Windows installer · Download the Visual Studio ...

Docs Download Visual Studio Code. Download VS Code for ...

- ເປົ້າ Web Browser ກຸຽນ visual studio code ແລ້ວດາວໂຫຼດ Download



05

Vscode setup for C



- กดเลือก Download ตาม OS ที่ใช้งาน
ตามขั้นตอนที่แนะนำ ติดตั้งให้เรียบร้อย

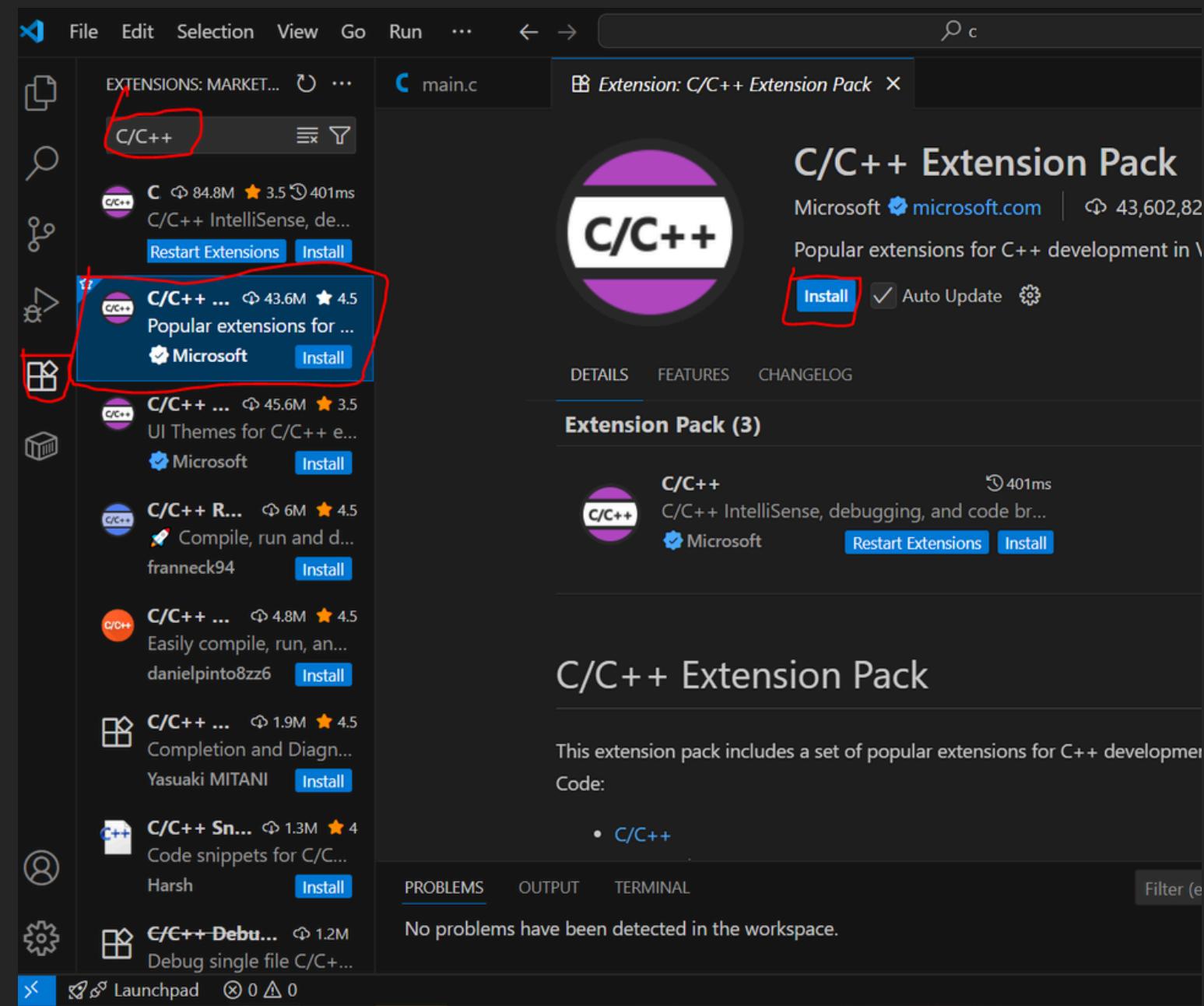


Search



05

Vscode setup for C

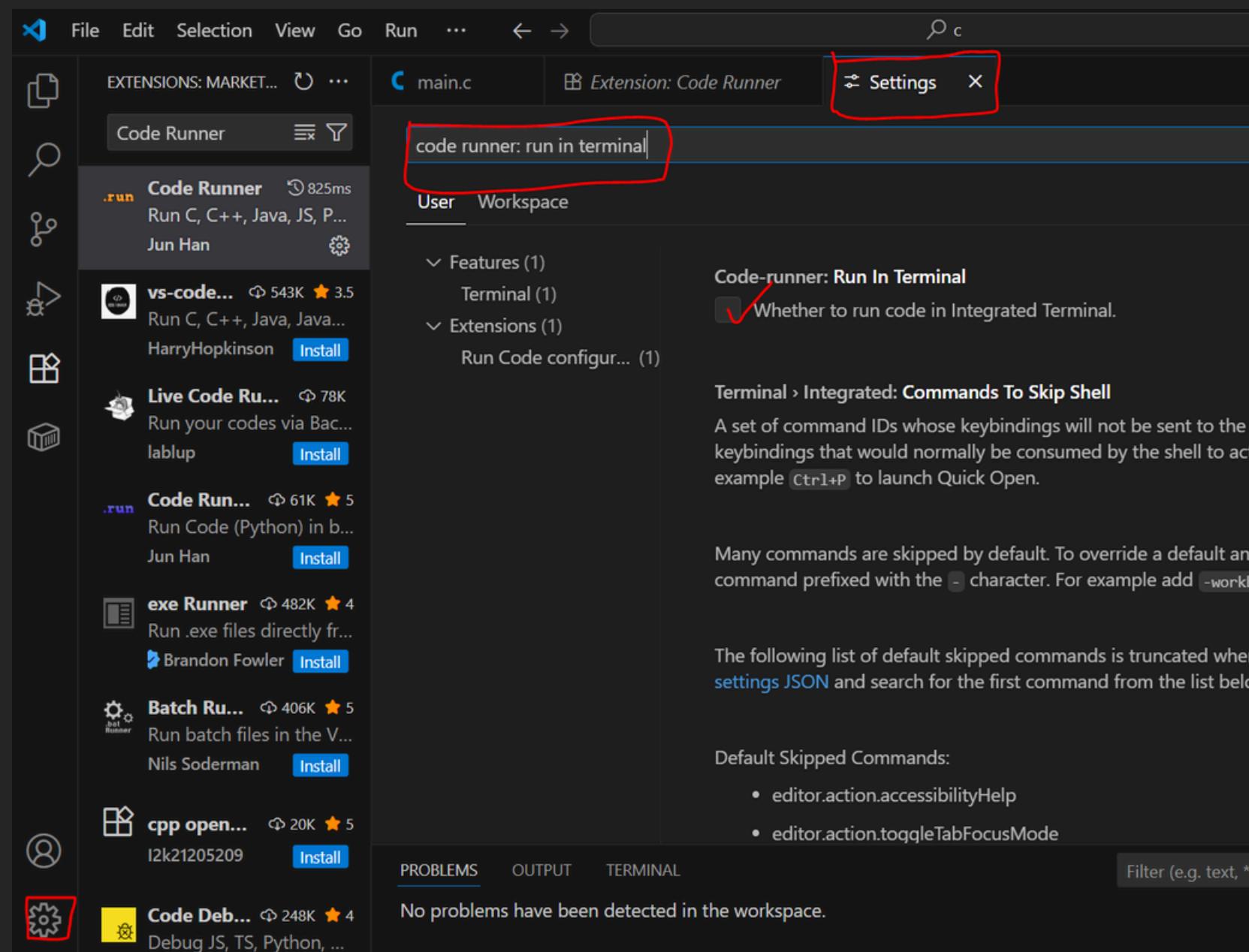


- ໃຫ້ Visual Studio Code ນາຕາມກົງເພື່ອຕັດຕັ້ງ C/C++ Extension Pack



05

Vscode setup for C

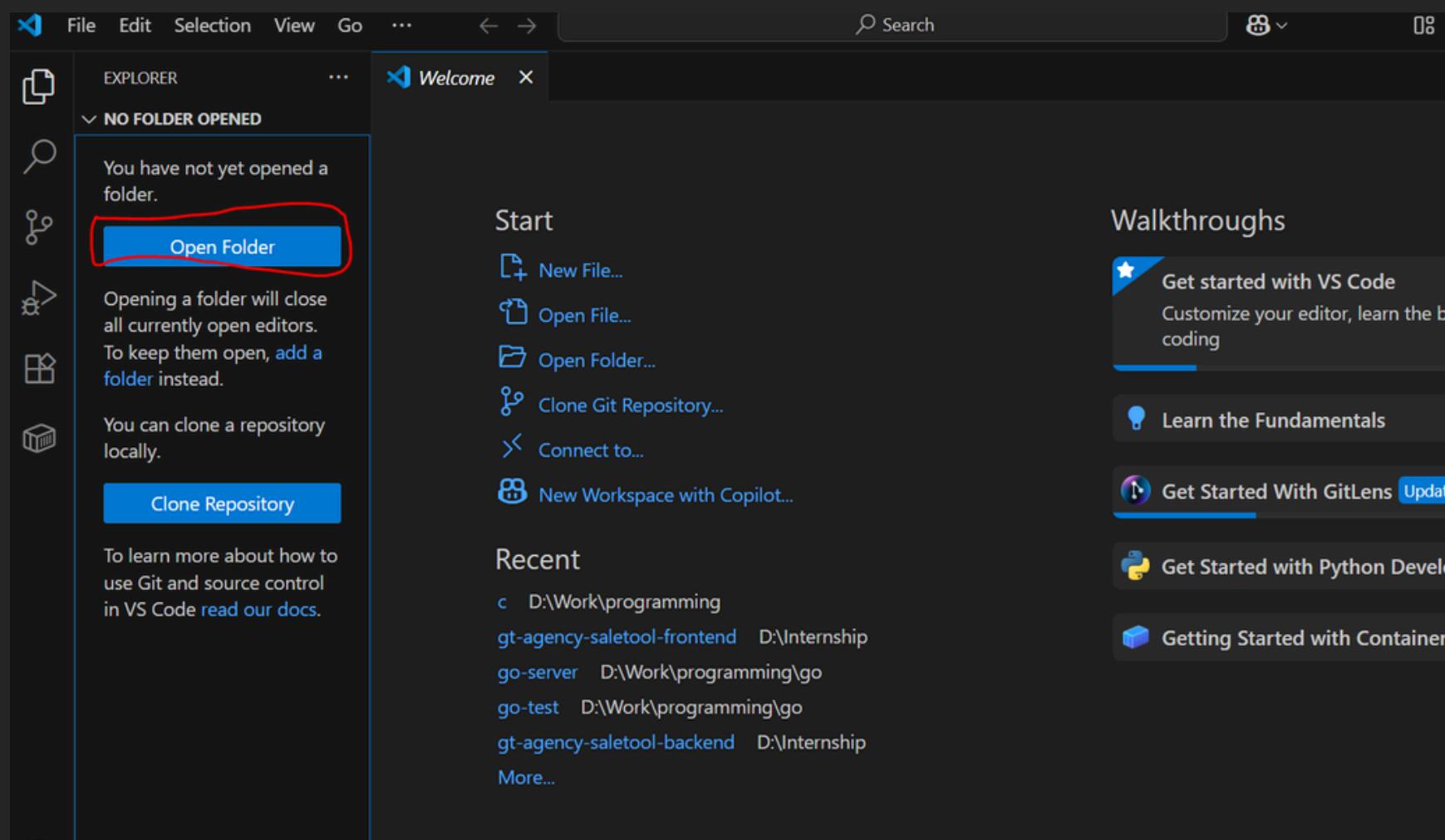


- გა მოიწოდეთ და გვიხვით **Settings** ჯგუფი
კურსი **code runner: run in terminal** განასაზღვროთ



05

Vscode setup for C



- **ជាត់ន open folder តើរូបសេខ**



05

Vscode setup for C

The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar, which has a folder named 'c' expanded, showing a file named 'main.c'. A red box highlights the 'main.c' file in the Explorer. The main area is the Editor, showing the code for a 'Hello, World!' program:

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Hello, World!\n");
5     return 0;
6 }
```

- នូវសរាយ file main.c ដោយបិទ code តាមចំណាំ



05

Vscode setup for C

The screenshot shows the Visual Studio Code interface. In the Explorer sidebar, there is a folder named 'c' containing 'main.c' and 'main.exe'. The main editor tab shows the following C code:

```
#include <stdio.h>
int main() {
    printf("Hello, World!\n");
    return 0;
}
```

In the bottom right corner of the editor, there is a red play button icon. Below the editor, the terminal tab is active, displaying the command line output:

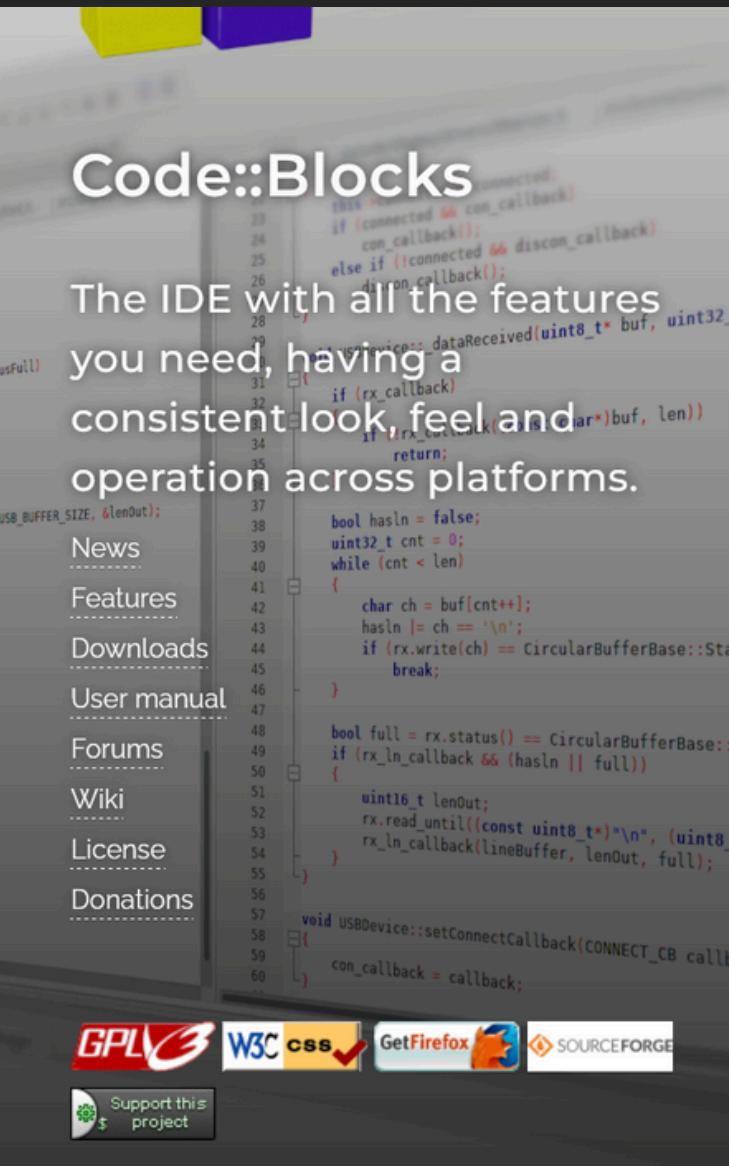
```
PS D:\Work\programming\c> cd "d:\Work\programming\c\" ; if ($?) { gcc main.c -o main } ; if ($?) { ./main }
Hello, World!
PS D:\Work\programming\c>
```

- កណ្តូរសម្រេចឡើងដើរ run code ឬផ្លូវតាម terminal ដោយលាង
- តម្រូវការ set up



05

Code::Blocks for C



Code::Blocks / Downloads / Binary releases

Binary releases

Please select a setup package depending on your platform:

- Windows XP / Vista / 7 / 8.x / 10
- Linux 32 and 64-bit
- Mac OS X

NOTE: For older OSes use older releases. There are releases for many OS version and platforms on the [Sourceforge.net](#) page.

NOTE: There are also more recent nightly builds available in the [forums](#) or (for Ubuntu users) in the [Ubuntu PPA repository](#). Please note that we consider nightly builds to be stable, usually.

NOTE: We have a [Changelog](#) for 25.03, that gives you an overview over the enhancements and fixes we have put in the new release.

NOTE: The default builds are 64 bit (starting with release 20.03). We also provide 32bit builds only for convenience.

Microsoft Windows (64 bit, default)

File

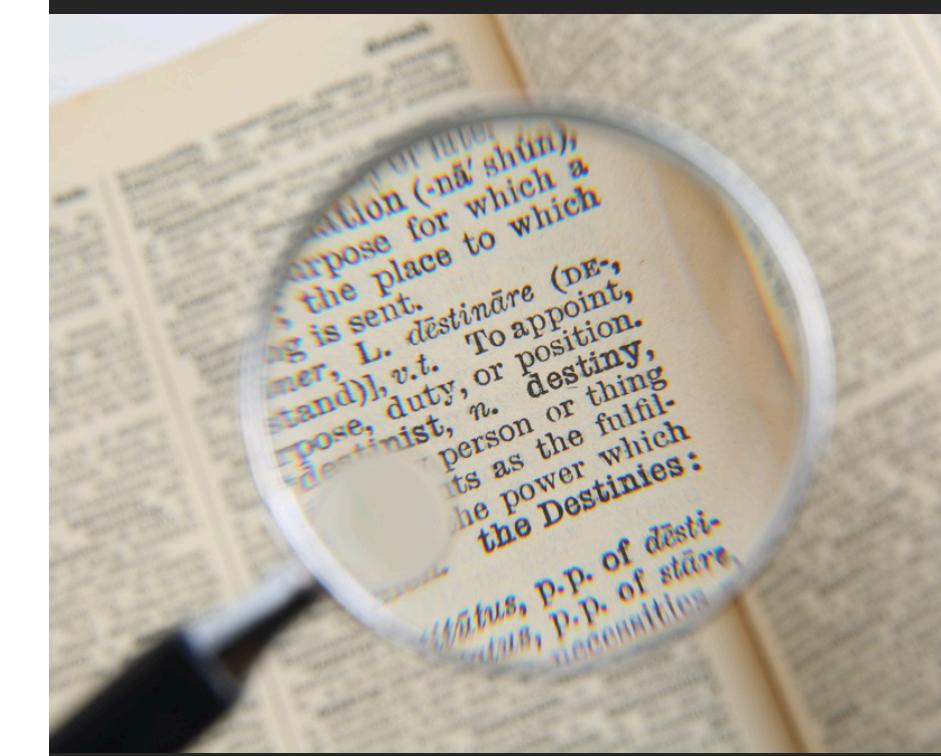
codeblocks-25.03-setup.exe

codeblocks-25.03-setup-nonadmin.exe

Download from

Sourceforge.net

Sourceforge.net

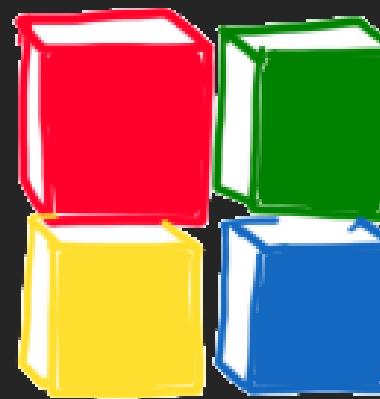


Google



05

Code::Blocks for C



Code::Blocks

The screenshot shows the SourceForge website with the URL [Home / Open Source Software / Software Development / Integrated Development Environments \(IDE\) / Code::Blocks](https://sourceforge.net/projects/codeblocks/). The page features the Code::Blocks logo, a brief description as 'A free C, C++ and Fortran IDE', and credits to 'Brought to you by: killerbot, mandrav, mortenmacfly, thomas-denk'. A prominent button says 'Your download will start shortly...'. Below it, there's a form to 'Get an email when there's a new version of Code::Blocks' with fields for 'Enter your email address' and 'Next'. Navigation links include 'Share This' and 'Problems Downloading?'. The top navigation bar includes links for Business Software, Open Source Software, SourceForge Podcast, Resources, and For Vendors.

Code::Blocks หรือ CodeBlocks เป็นโปรแกรม IDE สำหรับเขียนโปรแกรมภาษา C หรือ C++ โดยตัวโปรแกรม CodeBlocks เขียนขึ้นมาจากภาษา C++ ตัวโปรแกรมนี้เปิดเป็น Open-source

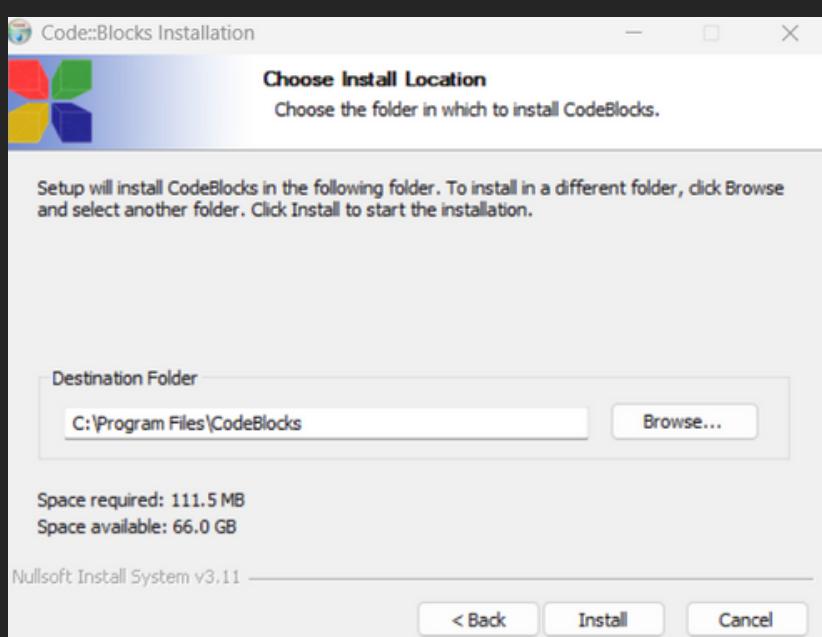
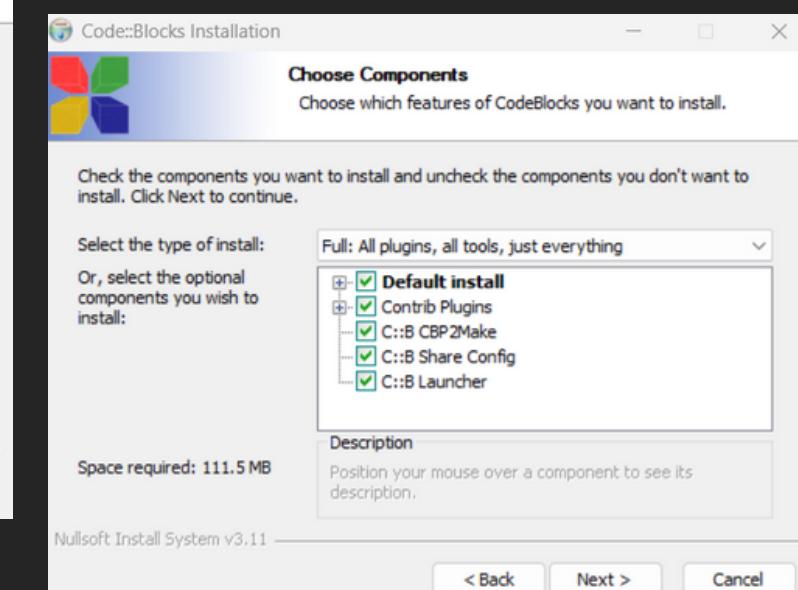
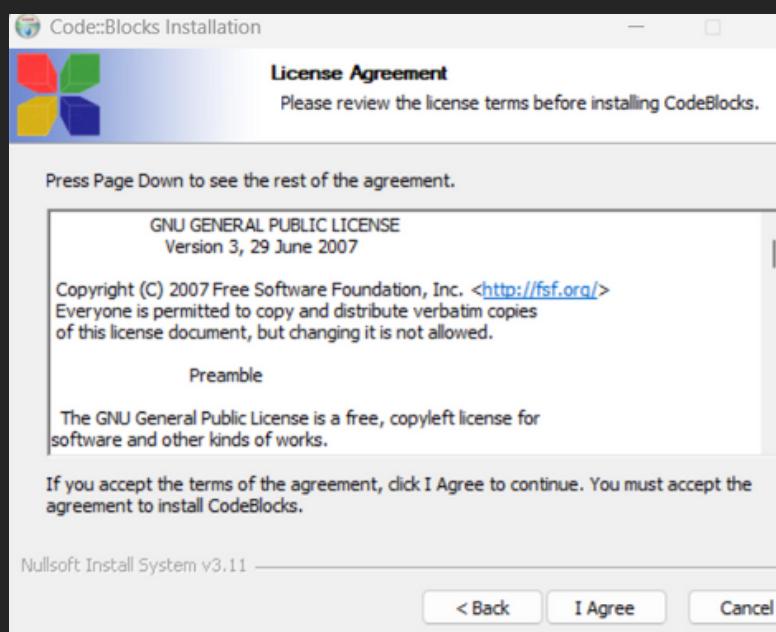
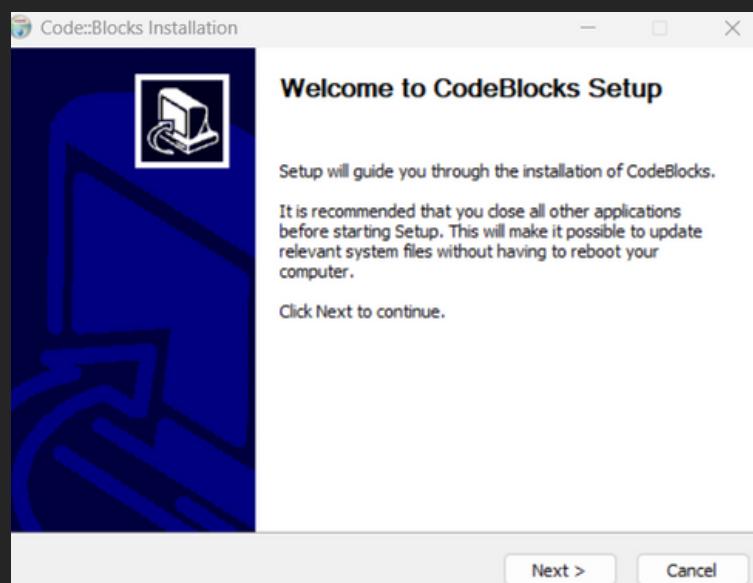


CodeBlock: Sourceforce
download application

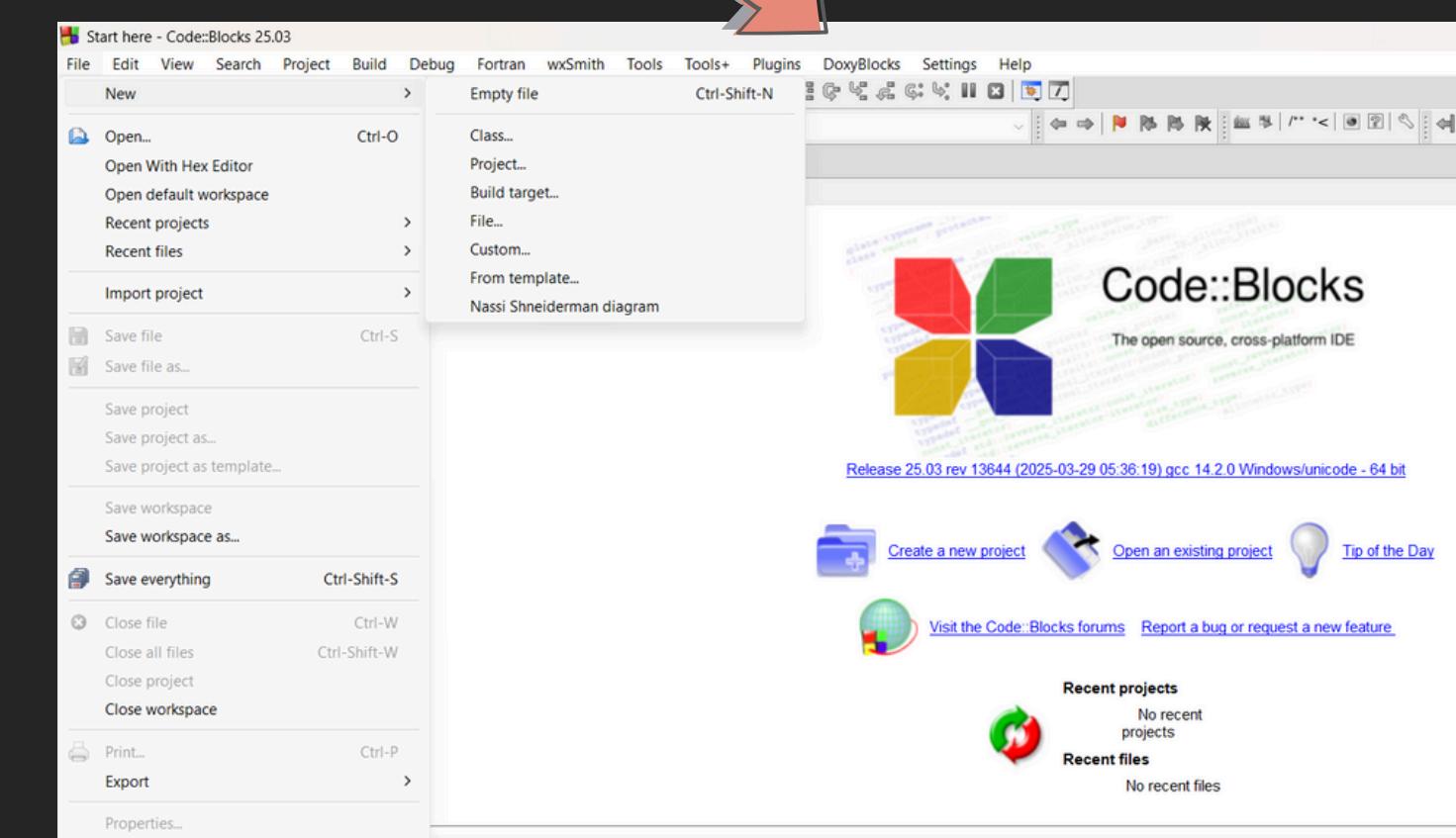
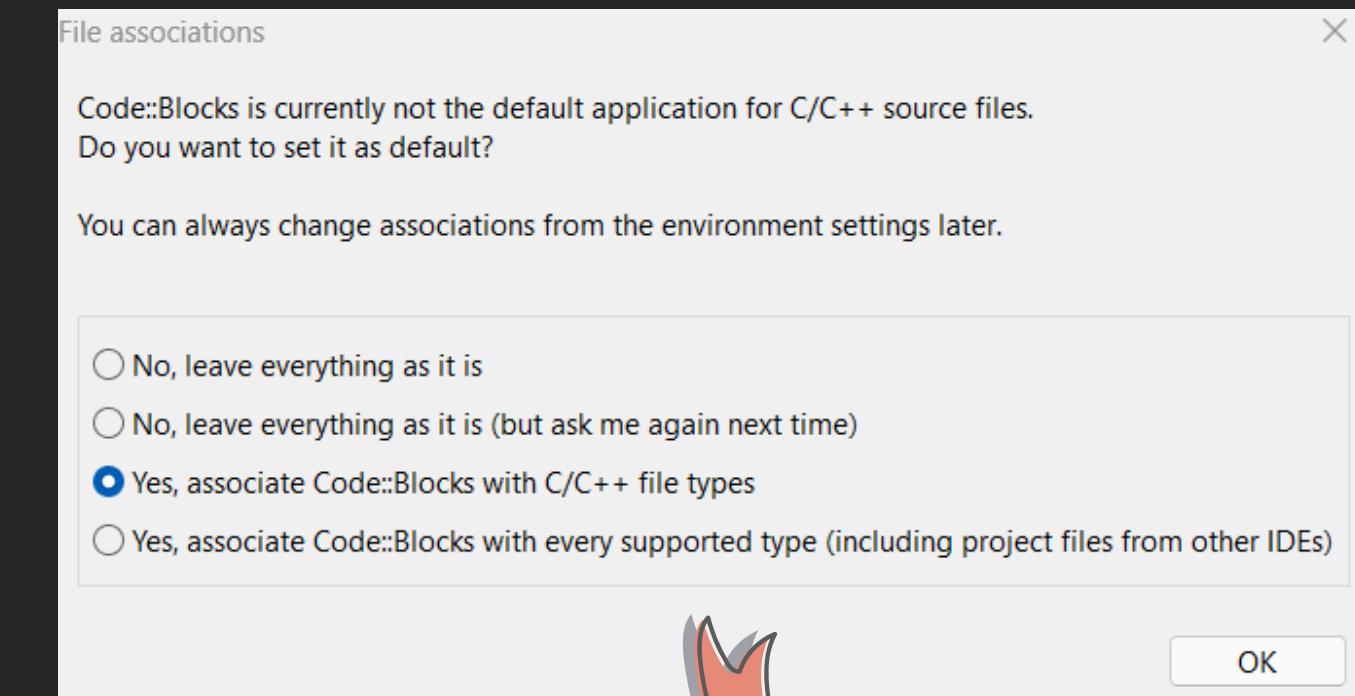
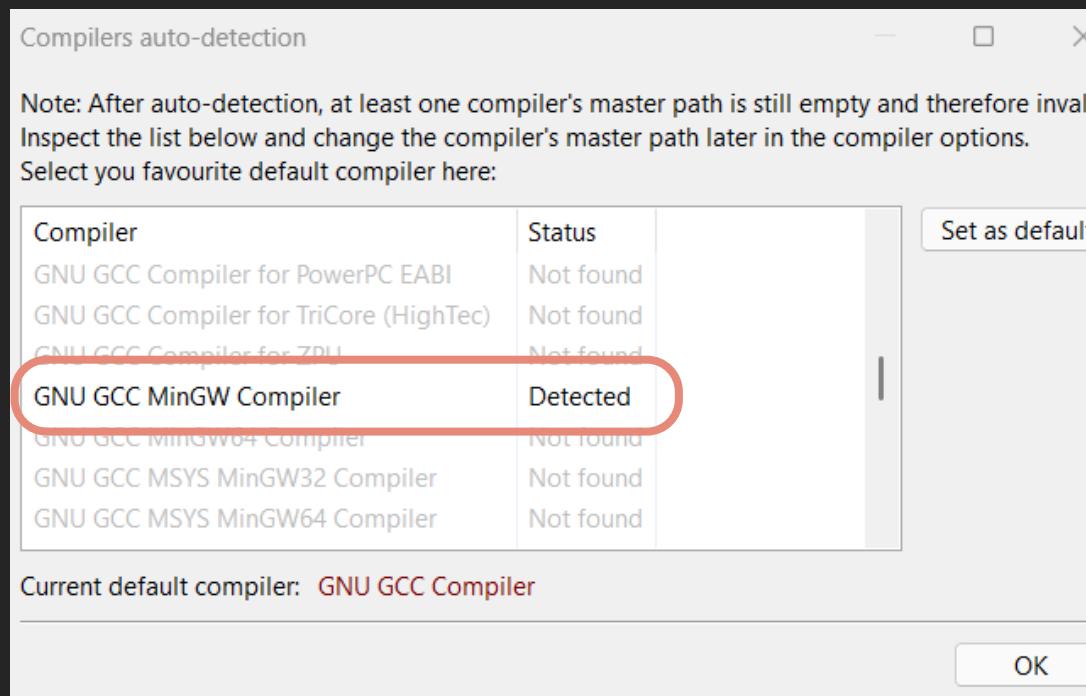


05

Code::Blocks for C



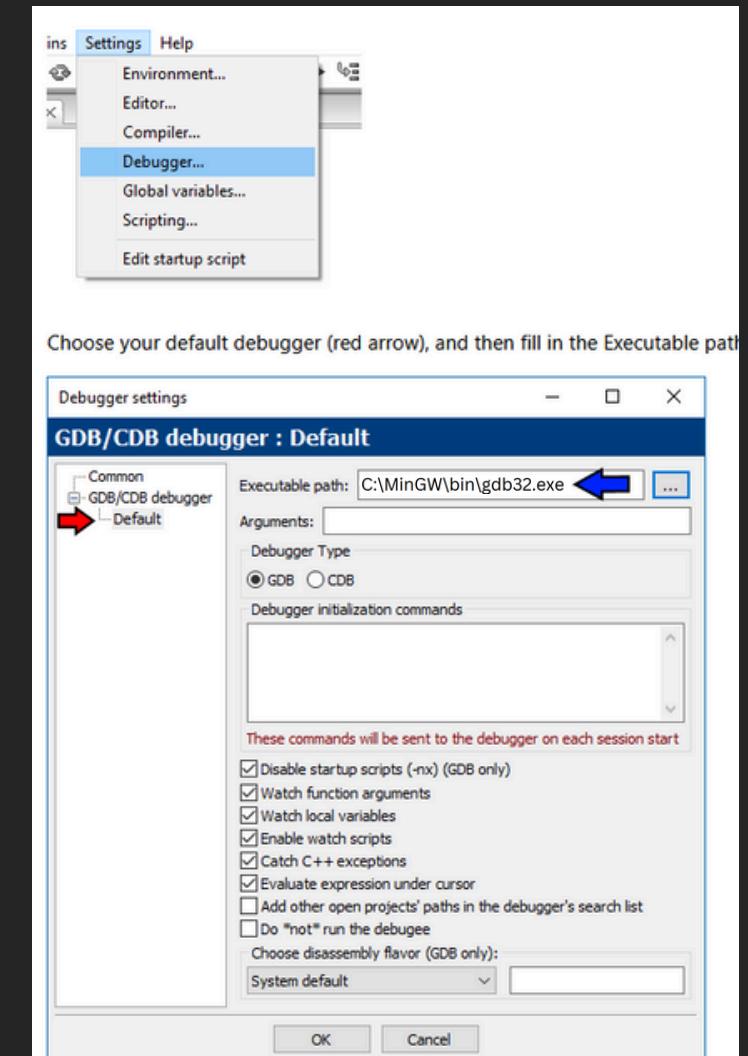
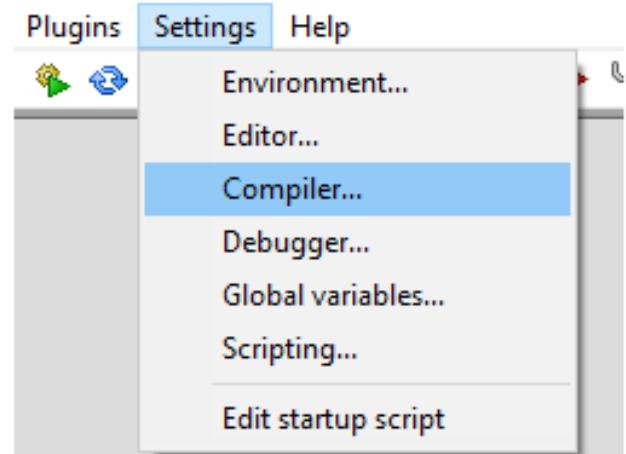
Code::Blocks (On Application)



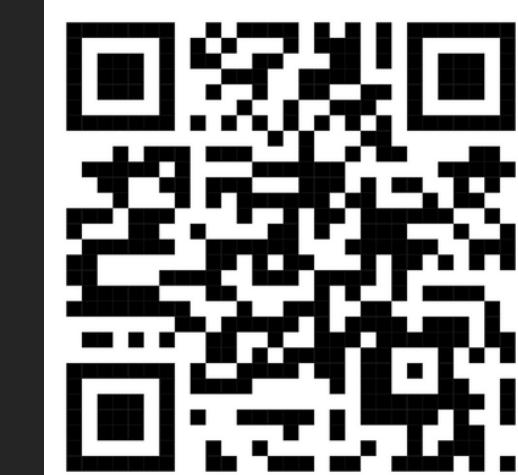
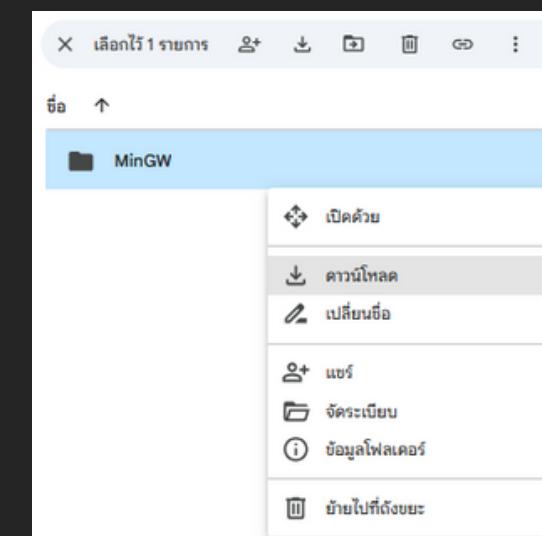
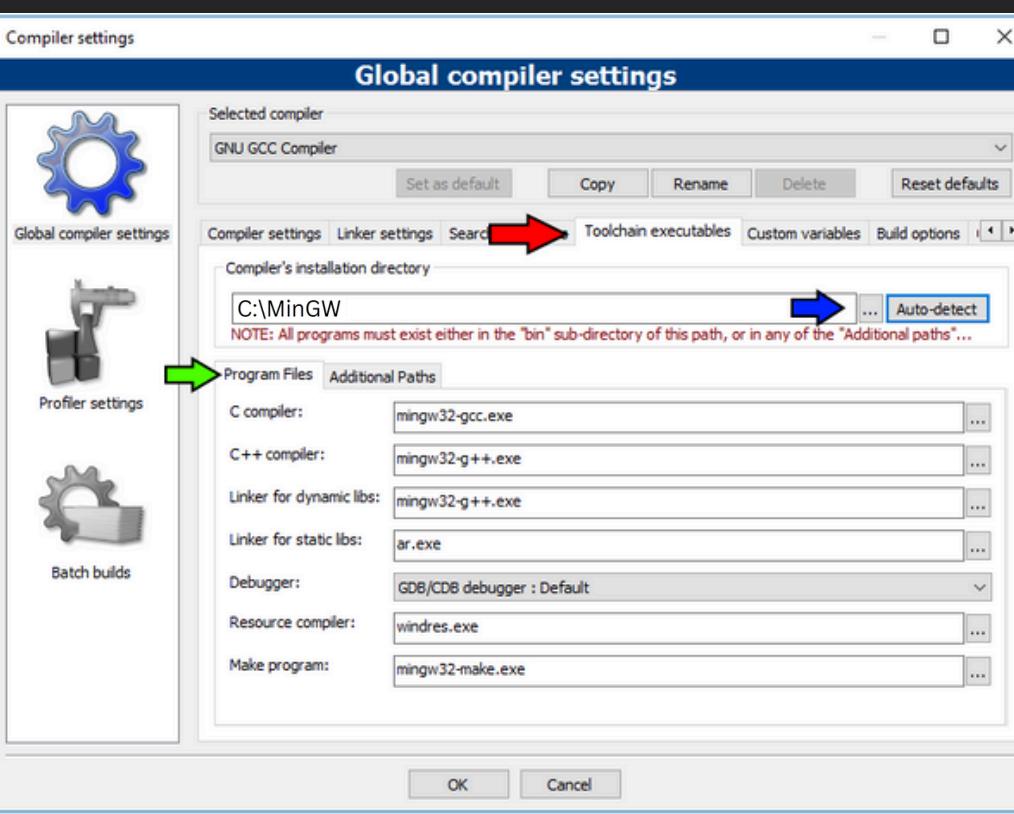
Code::Blocks setup MinGW

Code::Blocks Configuration

Go to your Compiler settings:

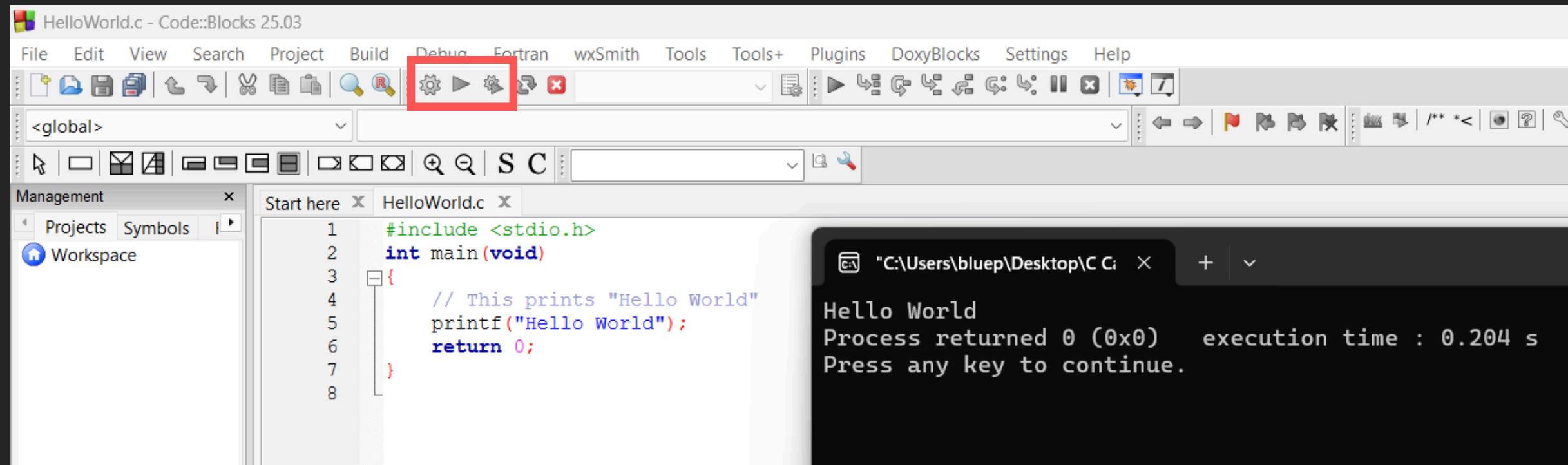


Download เป็น Zip file
แล้วเอาไปลงใน path folder



Download MinGW: Google Drive

Code::Blocks how to use



HelloWorld.c - Code::Blocks 25.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

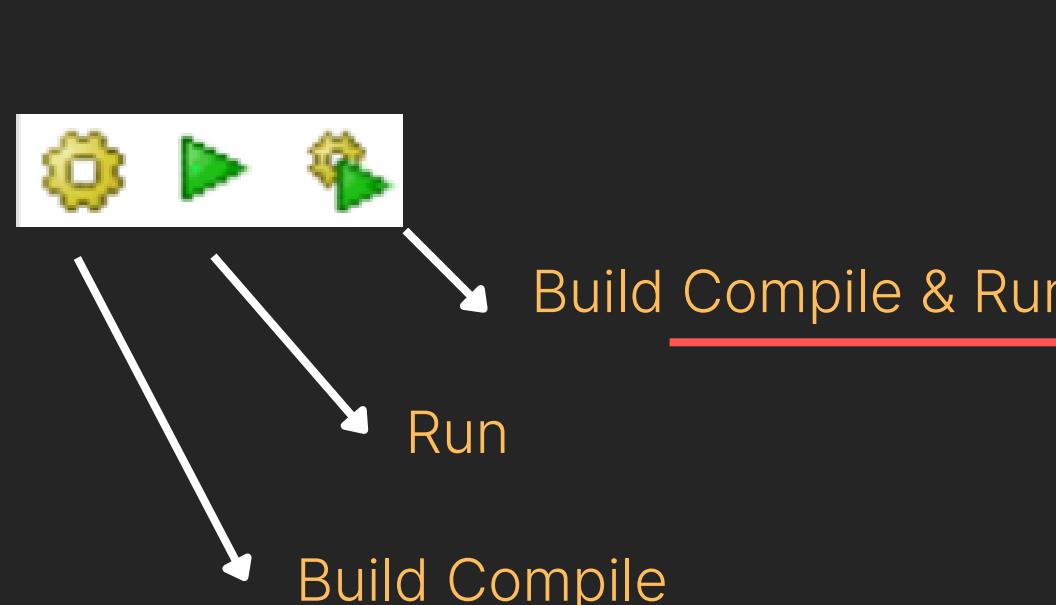
Management x Start here x HelloWorld.c x

```
1 #include <stdio.h>
2 int main(void)
3 {
4     // This prints "Hello World"
5     printf("Hello World");
6     return 0;
7 }
```

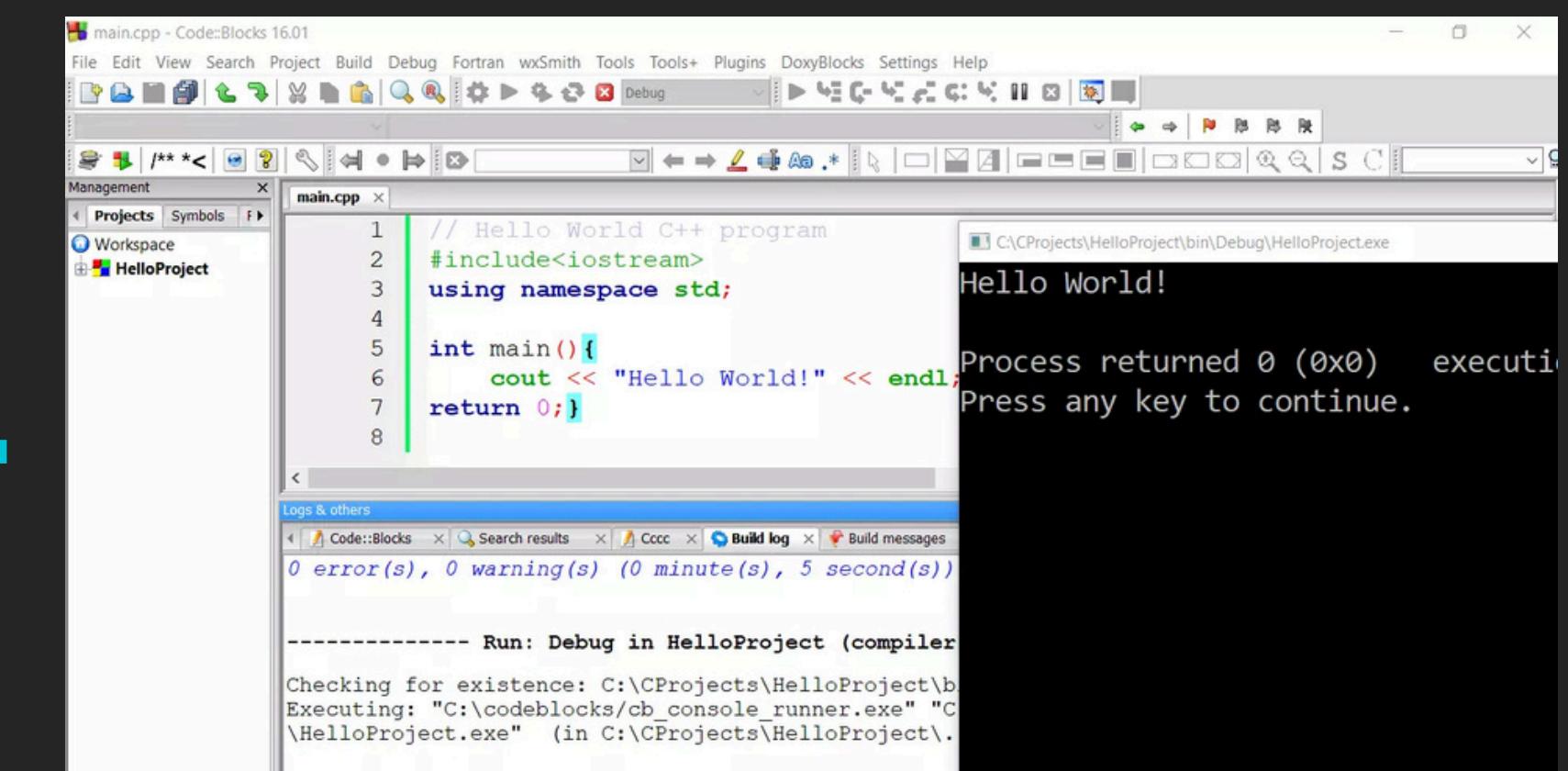
C:\Users\bluep\Desktop\C C x + v

```
Hello World
Process returned 0 (0x0) execution time : 0.204 s
Press any key to continue.
```

C



C++



main.cpp - Code::Blocks 16.01

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Management x Projects Symbols F x

Workspace HelloProject

```
1 // Hello World C++ program
2 #include<iostream>
3 using namespace std;
4
5 int main(){
6     cout << "Hello World!" << endl;
7     return 0;
8 }
```

C:\CProjects\HelloProject\bin\Debug\HelloProject.exe

```
Hello World!
Process returned 0 (0x0) execution time : 0.000 s
Press any key to continue.
```

Logs & others

0 error(s), 0 warning(s) (0 minute(s), 5 second(s))

----- Run: Debug in HelloProject (compiler)

Checking for existence: C:\CProjects\HelloProject\b

Executing: "C:\codeblocks\cb_console_runner.exe" "C\HelloProject.exe" (in C:\CProjects\HelloProject\.)



Search

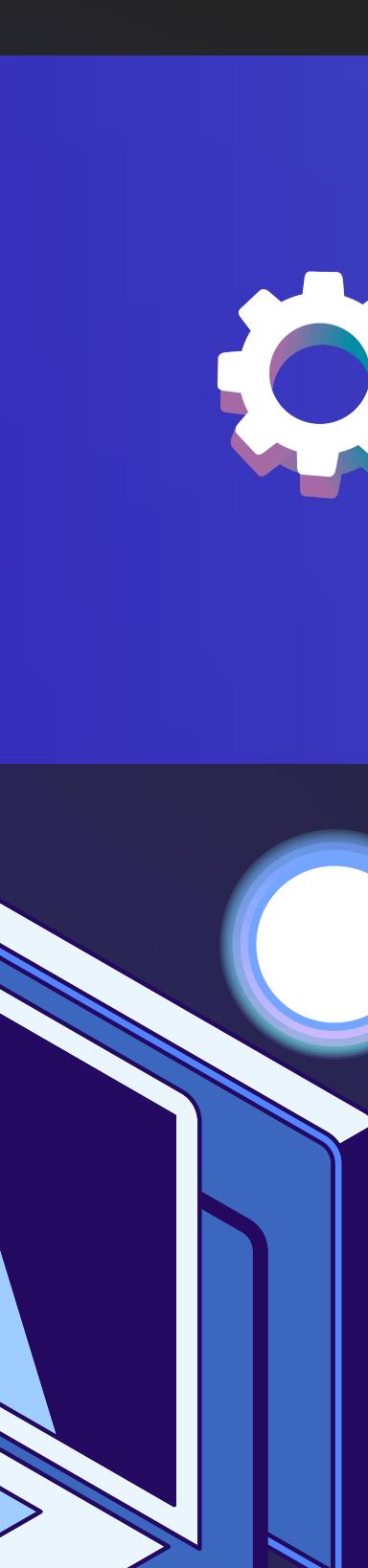
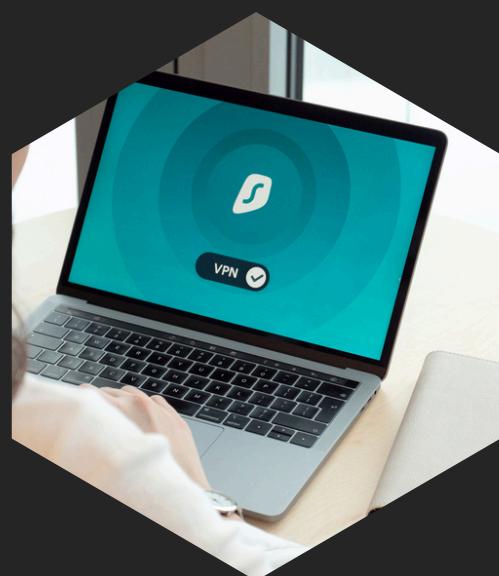
Let's Take a Break 5 Minutes!

Breakslides



CODEBOTIX

PROGRAM SYNTAX



```
main.c
```



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

`#include <stdio.h>` เป็นคำสั่ง Preprocessor Directive กี่ใช้สำหรับรวมไฟล์ header โดยที่ stdio.h ย่อมาจาก Standard Input/Output ช่วยให้สามารถใช้ฟังก์ชัน printf() ได้

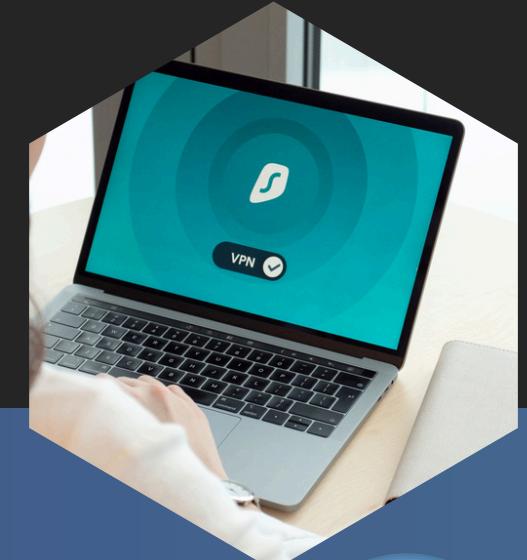
`int main()` เป็น Main Function หรือฟังก์ชันหลัก จุดเริ่มต้นของการทำงานของโปรแกรม โดยที่ int หมายถึงฟังก์ชันนี้จะคืนค่าเป็นตัวเลขจำนวนเต็ม

`printf("Hello World");` เป็นฟังก์ชันสำหรับแสดงผลoutput หน้าจอ ข้อความที่จะแสดงต้องอยู่ภายในเครื่องหมายคำพูด """

`return 0;` ส่งค่า 0 กลับไปยังระบบปฏิบัติการ 0 หมายถึงโปรแกรมทำงานสำเร็จ ค่าอื่นๆ (เช่น 1, -1) หมายถึงเกิดข้อผิดพลาด



BASIC C/C++ LANGUAGE



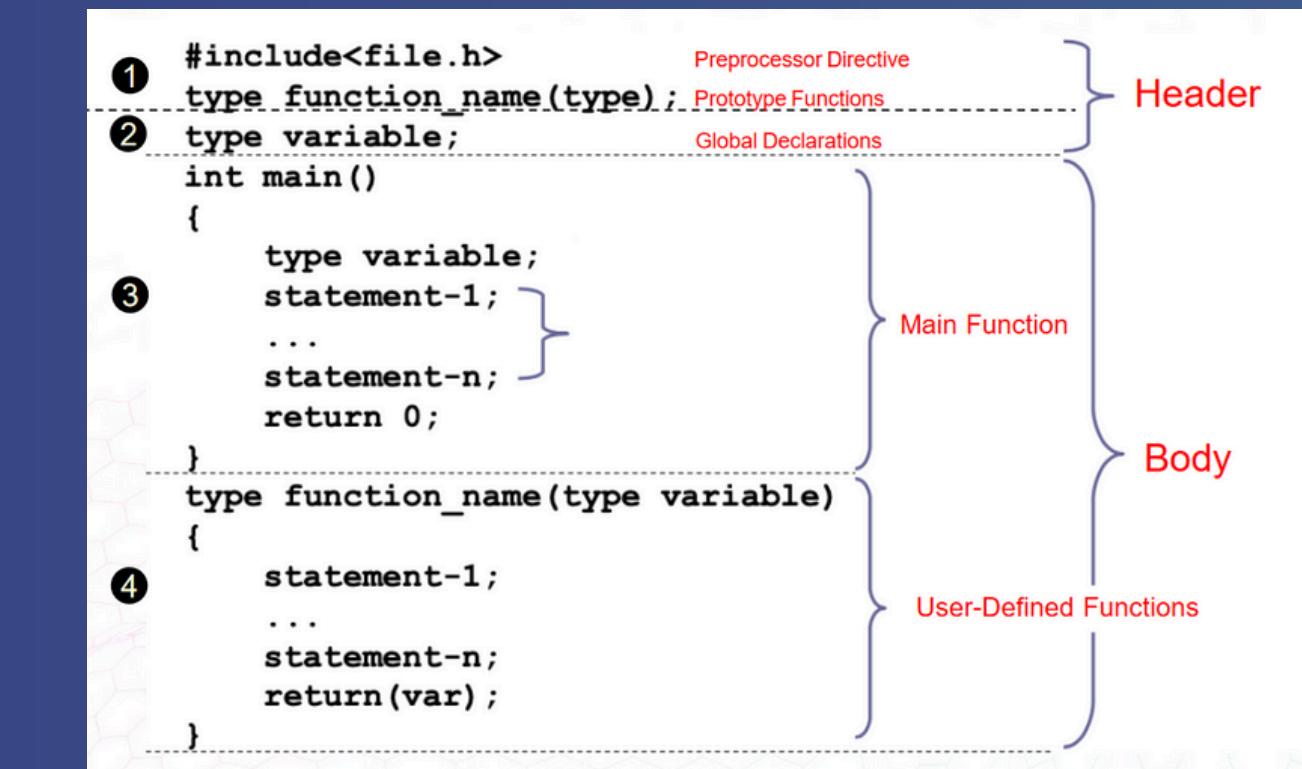
การประกาศและให้ค่าให้กับตัวแปร

สามารถประกาศได้ 2 วิธี

1. #define a 1

2. การประกาศตัวแปรใน C++ ทำได้ด้วยการระบุชนิดของตัวแปรและชื่อของตัวแปร

```
#include  
  
int  
float  
char  
string  
unsigned  
  
int main() {  
    // ประกาศตัวแปรชนิด int  
    int my_number;  
  
    // ให้ค่าให้กับตัวแปร  
    my_number = 42;  
    return 0;  
}
```



Link : เอกสารอ้างอิงอ่านต่อได้ที่ <https://kongruksiam.medium.com/พื้นฐานการเขียนโปรแกรมภาษา-c-ฉบับย่อ-bb1f3bec80b8>

https://www.compro.rmutt.ac.th/wp-content/uploads/2020/08/CH3_Introduction_to_C-1.pdf



07

การแสดงผลด้วย printf และ การรับค่าทางคีย์บอร์ดด้วย scanf

ฟังก์ชัน printf

```
printf(format-string, arg...);
```

1. format-string

คือ ข้อความหรือรูปแบบที่ต้องการให้แสดงบนหน้าจอ

- **ข้อความธรรมดา เช่น "Hello World"**
- **Format Specifier เช่น %d, %f, %c สำหรับแสดงค่าตัวแปร**



ฟังก์ชัน printf

• format specifier

ชนิดข้อมูล	format specifier	ตัวอย่างค่า	หมายเหตุ
จำนวนเต็ม (int)	%d	25	แสดงตัวเลขจำนวนเต็ม
ทศนิยม (float)	%f	3.14	แสดงตัวเลขทศนิยม
อักขระ (char)	%c	'A'	แสดงตัวอักษรตัวเดียว
ข้อความ (string)	%s	"Hello"	ใช้กับ char array เช่น <code>char name[50];</code>

ฟังก์ชัน printf

2. data-list

คือ ค่าที่ต้องการนำไปแสดงผล โดยจะจับคู่กับ format specifier ตามลำดับที่ปรากฏใน format-string

ตัวอย่าง

```
printf("Hello, my name is %s. I am %d years old.", "Lamine", 17);
```



ฟังก์ชัน printf

ตัวอย่างการใช้งาน ฟังก์ชัน printf

การใช้งาน printf ทุกครั้งต้องมีการ include header file จาก stdio.h ก่อนเสมอ

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

ตัวอย่างที่ 1: พิมพ์ข้อความธรรมดា

```
printf("Hello, World!");
```

ผลลัพธ์

```
Hello, World!
```

ຟົງກໍບັນ printf

ຕັວອຍ່າງທີ 2: ໃຫ້ format specifier ແສດອງຕັ້ງແປສ

```
int age = 20;  
printf("You are %d year(s) old.", age);
```

ຜາລັພຣ

```
You are 20 year(s) old.
```

ฟังก์ชัน printf

ตัวอย่างที่ 3: แสดงค่าหลายตัว

```
char name[] = "Anna";
int age = 18;
float gpa = 3.75;

printf("Name: %s\nAge: %d\nGPA: %.2f\n", name, age, gpa);
```

ผลลัพธ์

```
Name: Anna
Age: 18
GPA: 3.75
```



ฟังก์ชัน printf

Escape Character

คือ ตัวอักษรพิเศษที่ขึ้นต้นด้วยเครื่องหมาย \ (backslash) เพื่อบอกให้คอมพิวเตอร์
เข้าใจว่า "ข้อความถัดไปไม่ใช่ตัวอักษรธรรมดา แต่มีความหมายพิเศษ"

Escape Character	ความหมาย	ตัวอย่างใน printf()	ผลลัพธ์ที่ได้
\n	ขึ้นบรรทัดใหม่ (new line)	printf("Hello\nWorld");	Hello World
\t	เคาะ tab (ระยะห่างพิเศษ)	printf("Name\tAge");	Name Age
\\"	แสดง \"	printf("C:\\\\Program Files");	C:\\Program Files
\\"	แสดง \" (เครื่องหมายคำพูด)	printf("He said: \\\"Hi!\\\"");	He said: "Hi!"
\\'	แสดง ' (single quote)	printf("It\\'s OK.");	It's OK.



คำามชวนคิด

ผลลัพธ์ที่ได้จากโปรแกรมนี้แสดงผลทางหน้าจอจะเป็นอย่างไร ?

```
#include<stdio.h>

int main() {
    int A, B, C;
    A = 20;
    B = 30;
    C = A + B;

    printf("%d + %d = %d", A, B, C);

    return 0;
}
```

A. **%d + %d = %d**

B. **A + B = C**

C. **20 + 30 = 50**

D. **20 + 30 = 20 + 30**



ฟังก์ชัน `scanf`

เป็นฟังก์ชันสำหรับ รับค่าจากผู้ใช้ทางคีย์บอร์ด

```
scanf(format-string, arg...);
```

ตัวอย่างที่ 1: รับค่าอายุ (int)

```
int age;
printf("Please enter your age: ");
scanf("%d", &age);
printf("You are %d year(s) old.", age);
```

ຟົງກໍບັນ `scanf`

ຕັວອຍ່າງທີ່ 2: ຮັບຄໍາເກສາເລື່ອຍ (float)

```
float gpa;  
printf("Please enter your GPA: ");  
scanf("%f", &gpa);  
printf("Your GPA is %.2f\n", gpa);
```



ฟังก์ชัน scanf

ตัวอย่างที่ 3: รับอักขระ (char)

```
char c;  
printf("Enter a character: ");  
scanf("%c", &c);  
printf("You entered: %c", c);
```



ຟົງກໍບັນ `scanf`

ຕັວອຍ່າງທີ 4: ຮັບຂ້ອຄວາມ (string)

```
char name[50];
printf("Please enter your name: ");
scanf("%s", name);
printf("Welcome, %s\n", name);
```



ພົນກໍບັນ `scanf`

ຕົວຢ່າງທີ 5: ສັບຄ່າ `int`, `float` ແລະ `char` ດ້ວຍ `scanf()` ເລີຍວ

```
int age;
float gpa;
char grade;

printf("Enter your information: Age GPA Grade (e.g., 18 3.50 A):\n");
scanf("%d %f %c", &age, &gpa, &grade);

printf("\n===== The information you entered =====\n");
printf("Age: %d years\n", age);
printf("GPA: %.2f\n", gpa);
printf("Grade: %c\n", grade);
```

ฟังก์ชัน scanf

ตัวอย่างที่ 6: รับค่า string ที่มี whitespace

```
char str[100];
printf("Enter a string with spaces:\n");
scanf(" %[^\n]", str);
```



คำamuraนคิด

ພາລັພ່ນ printf ພສທິດສຸດທ້າຍເປັນວ່າງໄສ

```
#include<stdio.h>

int main() {
    char name[100];
    printf("Please enter your name:");
    scanf("%s", name);

    printf("You entered: %s", name);
    return 0;
}
```

Input:

Please enter your name: Lamine Yamal

Output:

A. You entered: Lamine

B. You entered: Lamine Yamal

Operators in C

1. ตัวดำเนินการ (Operators) คืออะไร?

ตัวดำเนินการ (Operators) ในภาษา C คือ สัญลักษณ์ที่ใช้สำหรับดำเนินการต่างๆ กับตัวแปรและค่าคงที่ เช่น การคำนวณทางคณิตศาสตร์ การเปรียบเทียบ การกำหนดค่า เป็นต้น

Operators in C

2. ประเภทของตัวดำเนินการในภาษา C

- **ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operators)**

```
int a = 10, b = 3;

printf("%d\n", a + b);    // บวก (13)
printf("%d\n", a - b);    // ลบ (7)
printf("%d\n", a * b);    // คูณ (30)
printf("%d\n", a / b);    // หาร (3) - หารแบบจำนวนเต็ม
printf("%d\n", a % b);    // หารเอาเศษ (1)
printf("%d\n", ++a);      // เพิ่มค่าก่อนใช้ (11)
printf("%d\n", b--);      // ลดค่าหลังใช้ (3) และ b จะเป็น 2
```

Operators in C

- ตัวดำเนินการกำหนดค่า (Assignment Operators)

```
int x = 5;      // กำหนดค่า
x += 3;        // เท่ากับ x = x + 3 (8)
x -= 2;        // เท่ากับ x = x - 2 (6)
x *= 4;        // เท่ากับ x = x * 4 (24)
x /= 3;        // เท่ากับ x = x / 3 (8)
x %= 5;        // เท่ากับ x = x % 5 (3)
```



Operators in C

- ตัวดำเนินการเปรียบเทียบ (Relational Operators)

```
int m = 5, n = 7;

printf("%d\n", m == n); // เท่ากับ (0 - false)
printf("%d\n", m != n); // ไม่เท่ากับ (1 - true)
printf("%d\n", m > n); // มากกว่า (0)
printf("%d\n", m < n); // น้อยกว่า (1)
printf("%d\n", m >= n); // มากกว่าหรือเท่ากับ (0)
printf("%d\n", m <= n); // น้อยกว่าหรือเท่ากับ (1)
```

Operators in C

- ຕັ້ງດຳເນີນການຕະຫຼອນ: (Logical Operators)

```
int p = 1, q = 0;  
  
printf("%d\n", p && q); // AND (0)  
printf("%d\n", p || q); // OR (1)  
printf("%d\n", !p); // NOT (0)
```

Operators in C

- ตัวดำเนินการระดับบิต (Bitwise Operators)

```
unsigned int a = 5;      // 0101
unsigned int b = 3;      // 0011

printf("%d\n", a & b);  // AND ระดับบิต (1 - 0001)
printf("%d\n", a | b);  // OR ระดับบิต (7 - 0111)
printf("%d\n", a ^ b);  // XOR ระดับบิต (6 - 0110)
printf("%d\n", ~a);     // NOT ระดับบิต (ค่าจะขึ้นกับขนาดของ int)
printf("%d\n", a << 1); // เลื่อนบิตไปทางซ้าย (10 - 1010)
printf("%d\n", a >> 1); // เลื่อนบิตไปทางขวา (2 - 0010)
```

Operators in C

Operators Precedence

Category	Operator	Associativity
Postfix	<code>0 [] -> . ++ --</code>	Left to right
Unary	<code>+ - ! ~ ++ -- (type)* & sizeof</code>	Right to left
Multiplicative	<code>* / %</code>	Left to right
Additive	<code>+ -</code>	Left to right
Shift	<code><<>></code>	Left to right
Relational	<code><<= >>=</code>	Left to right
Equality	<code>== !=</code>	Left to right
Bitwise AND	<code>&</code>	Left to right
Bitwise XOR	<code>^</code>	Left to right
Bitwise OR	<code> </code>	Left to right
Logical AND	<code>&&</code>	Left to right

คำamuraคิด

ผลลัพธ์ printf เป็นอย่างไร ?

```
#include <stdio.h>
int main() {
    printf("%d", (40/4*3+4*5/2));
    return 0;
}
```

40



Search

Let's Take a Break 5 Minutes!

Breakslides



CODEBOTIX

Condition

- **What is Condition in C ?**

"เงื่อนไข" หมายถึงนิพจน์ที่ประเมินผลเป็นจริง (ไม่ใช่คูณย์) หรือเป็นเท็จ (คูณย์) เงื่อนไขเหล่านี้มีความสำคัญพื้นฐานในการควบคุมการให้ของโปรแกรมผ่านคำสั่งการตัดสินใจ

- **If-else Conditional Statements**

- **Switch Case**

Condition

- **if-else เงื่อนไขแบบถ้า...นอกจานี้...**

```
int main() {
    int money;
    printf("Cash : ");
    scanf("%d", &money);

    if (money >= 200) {
        printf("จ่ายสำเร็จ");
    } else if (money >= 250) {
        printf("จ่ายแบบถ้วนของวงเงินเพิ่ม");
    } else {
        printf("ไม่สามารถทำรายการได้");
    }

    return 0;
}
```

✓ if เงื่อนไขถ้า

✓ else if เงื่อนไขรอง (ต้องมี if ก่อน)

✓ else เงื่อนไขนอกเหนือจากนี้

Condition

- **switch-case เงื่อนไขแบบ**

คำสั่ง switch case เป็นคำสั่งควบคุมเพื่อให้โปรแกรมทำงานแบบ
ถอยทางเลือกคล้ายกับคำสั่ง if else-if แต่การใช้งานของมันจะ^{เรียบง่าย} และ ^{จำกัดกว่ามาก} และในการกำหนดเงื่อนไขจะเป็นการ
เปรียบเทียบความเท่ากันเท่านั้น นี่เป็นรูปแบบการใช้งานคำสั่ง
switch case ในภาษา C

Condition

- **switch-case**

✓ ตัวอย่าง coding

```
switch (input)
{
    case VALUE_1:
        // statements
        break;
    case VALUE_2:
        // statements
        break;
    case VALUE_3:
        // statements
        break;
    default:
        // statements
}
```

ค่าที่ 1 เงื่อนไข A
ค่าที่ 2 เงื่อนไข B
ค่าที่ 3 เงื่อนไข C

```
int n = 2;
switch (n) {
case 1:
    printf("One\n");
    break;
case 2:
    printf("Two\n");
    break;
case 3:
    printf("Three\n");
    break;
default:
    printf("Unknown\n");
}
return 0;
```

Loop

- **What is Loop in C ?**

การวนซ้ำ คือโค้ดองสร้างควบคุมที่ช่วยให้สามารถรับข้อมูลคำสั่ง (code block) ซ้ำๆ ได้หลายครั้งตามที่ต้องการ โดยมี 3 ประเภท คือ

- **for**
- **while**
- **do-while**

Loop

- **while (loop)**

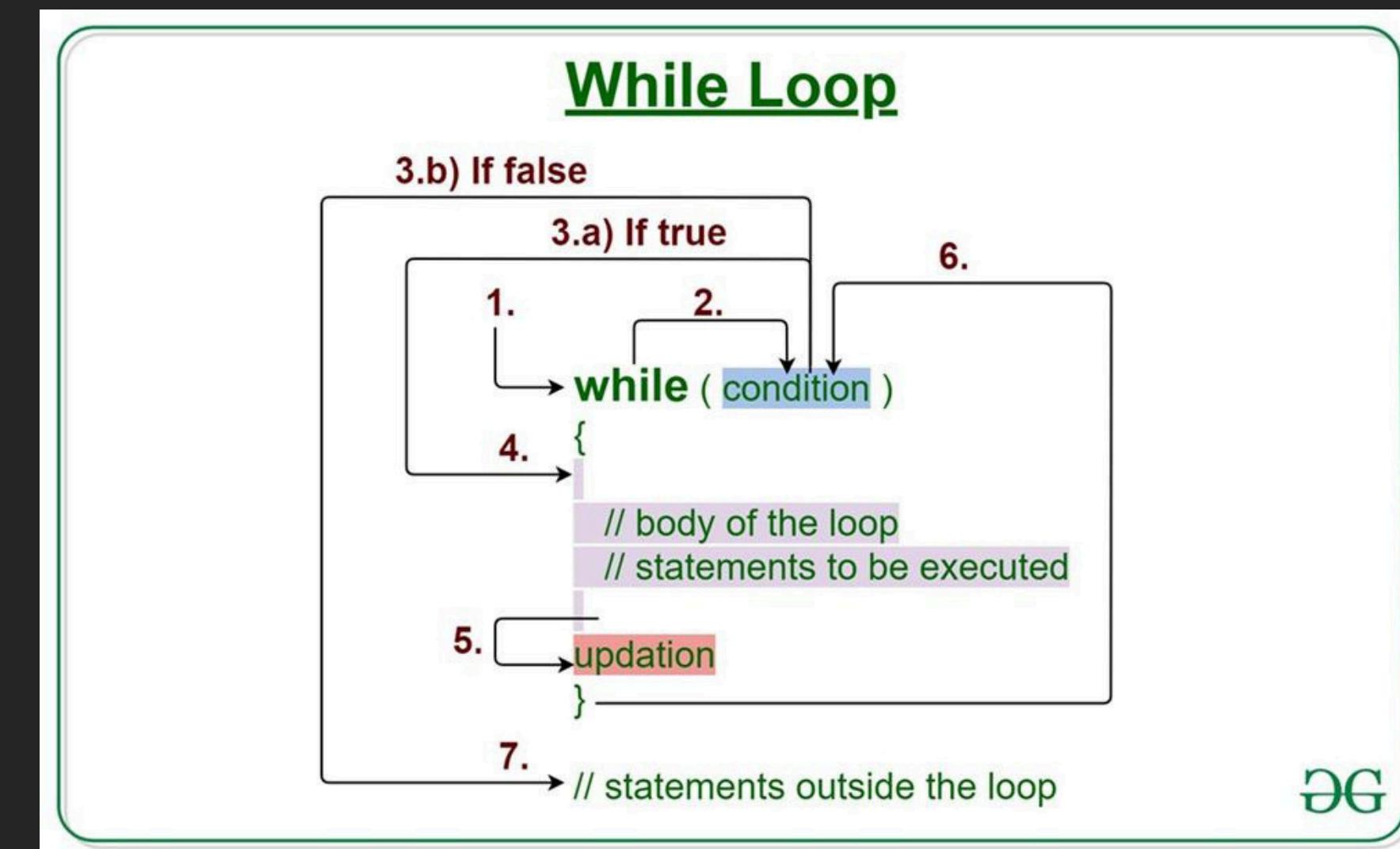
while loop เป็นคำสั่งวนซ้ำที่เป็นพื้นฐานและเรียบง่ายที่สุดในภาษา C มันใช้สำหรับควบคุมเพื่อให้โปรแกรมทำงานบางอย่างซ้ำๆ จนกว่าจะมีเงื่อนไขที่กำหนด โดยที่มีเงื่อนไข condition สำหรับลูปเพื่อกำหนดและส่วนของคำสั่งที่ต้องการให้ทำงานในขณะที่เงื่อนไขเป็นจริง

```
while ( i <= 10 ) {  
    (body)           condition is true  
    i++;  
}
```

```
#include <stdio.h>  
  
int main() {  
    int i = 1;  
  
    while (i <= 10) {  
        printf("%d ", i);  
        i++;  
    }  
  
    return 0;  
}
```

Loop

- **while (loop)**





Loop

• **for (loop)**

เป็นคำสั่งวนซ้ำที่ใช้สำหรับควบคุมให้โปรแกรมทำงานซ้ำๆ กายได้เงื่อนไขแต่มันถูกออกแบบมาสำหรับใช้กับการวนซ้ำในจำนวนรอบที่แน่นอน เมื่อจะจากเราสามารถกำหนดค่าเริ่มต้น เงื่อนไขและการเปลี่ยนแปลงค่าไว้ที่เดียวกัน นี่เป็นรูปแบบการใช้งานคำสั่ง for loop ในภาษา C

```
for ( i = 0; money >= 0 ; i++)  
    initial value           condition           end value
```



ถ้าไม่เคยประกาศตัวแปร `i` มา ก่อน ต้องใส่ `int` เพิ่มด้านหน้า

```
#include<stdio.h>
int main()
{
    int number; 1
    2 for(number=1;number<=10;number++)
    {
        printf("%d\n",number) 3
    }
    return 0;
}
```

Loop

- **do while (loop)**

ເໜືອນ while loop ແຕ່ໂປຣແກຣມຈະກຳຈານໃນລູບປ່ວຍ່າງນ້ອຍກ່ອນກັ່ງຮອບເສມວ
ນີ້ເປັນຮູບແບບການໃຊ້ຈານຄໍາສັ່ງ do-while loop ແລ້ວຄ່ອຍເຫັນ loop ໄວໃຊ້ກັບ
ກາສ setup ກີ່ຈຳເປັນເກື່ອງກັບ initial state

```
do
{
    // body of do...while loop
}
while (condition);
```

```
int main()
{
    int i=0;
    do
    {
        printf("while vs do-while\n");
    }while(i==1);

    printf("Out of loop");
}
```

Loop

- **break**

คำสั่ง break เป็นคำสั่งที่ใช้เพื่อควบคุมลูปให้จบการทำงานในทันที โดยที่ไม่สนใจว่าเงื่อนไขจะยังคงเป็นจริงอยู่หรือไม่ เราแค่ใช้มันร่วมกับคำสั่ง if เพื่อตรวจสอบเงื่อนไขก่อน เช่น **ถ้าหากเงื่อนไขบางอย่างเป็นจริง เราเรียกใช้คำสั่ง break เพื่อจบการทำงานของลูป**

```
for (int i = 1; i <= 10; i++) {  
    if (i == 6) {  
        break;  
    }  
    printf("%d ", i);  
}
```

i = 6 ตัดจบลูป

Loop

• continue

คำสั่ง **continue** ใช้สำหรับควบคุมเพื่อข้ามการทำงานของลูปในรอบปัจจุบันไปยังรอบใหม่ในทันที โดยที่โปรแกรมจะข้ามคำสั่งที่ปรากฏหลังจากคำสั่ง **continue** ทั้งหมด; **ไม่มีอบกับคำสั่ง break** คำสั่ง **continue** ยังคงทำงานในลูปในขณะที่เงื่อนไขเป็นจริงตามปกติ (แค่อ้างจะข้ามเลย ๆ)

```
for (int i = 1; i <= 10; i++) {  
    if (i % 2 == 1) {  
        continue;  
    }  
    printf("%d ", i);  
}
```

result:

2 4 6 8 10



ໂວກຍິ່ນການຫວັນ condition, Loop

• continue

- ເຊຍນໂປຣແກຣມຕັດເກສາ
 - ເຊຍນໂປຣແກຣມໜາພວອມແລບຄູ່ 1-100
 - ເຊຍນໂປຣແກຣມໜາຄ່າ factorial
 - ເຊຍນໂປຣແກຣມພິມພໍຽປດາວ (pattern)

result:

กรุณาใส่คะแนนสอบ (0-100): 85

คะแนน: 85, เกรด: A

Pattern 1: สามเหลี่ยมนูนจาก

* * * *

Pattern 3: สามเหลี่ยมหน้าจั่ว

*

Pattern 4: ข้าว荷กุมตัด

*

+
+

ผลรวมของเลขคี่ 1-100 = 2550

$$n! = n \times (n-1) \times (n-2) \times \dots \times 2 \times 1$$

$0! = 1$ **ans factorial**

กรุณาใส่ตัวเลข: 5

$$5! = 120$$

$$\text{จำนวน: } 5 \times 4 \times 3 \times 2 \times 1 = 120$$

ເລືອກວ່າຈະກຳ pattern ອະໄສ



Search

Let's Take a Break 8 Minutes!

Breakslides



CODEBOTIX



Ascii Table

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	,	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	-	127	7F	[DEL]

Ascii Table

ในการเขียนโปรแกรม C ตัวแปรอักขระจะเก็บค่า ASCII (จำนวนเต็มระหว่าง 0 ถึง 127) แทนที่จะเป็นอักขระนั้นเอง ค่าจำนวนเต็มนี้คือรหัส ASCII ของอักขระ

ตัวอย่างเช่น ค่า ASCII ของ 'A' คือ 65 และ 'a' คือ 97

ซึ่งหมายความว่า หากกำหนดให้ 'A' กับตัวแปรอักขระ 65 จะถูกเก็บไว้ในตัวแปรแทนที่จะอยู่ใน'A' ตัวตัวแปรเอง

Ascii Table

ตัวอย่าง การอ่านค่าตัวเลข Ascii สามารถเอาไปต่อ�่อ เช่น กด Space bar, Esc ถึงทำงาน

```
#include <stdio.h>
int main() {
    char c;
    printf("Enter a character: ");
    scanf("%c", &c);

    // %d displays the integer value of a character
    // %c displays the actual character
    printf("ASCII value of %c = %d", c, c);

    return 0;
}
```



Array

อาร์เรย์ คือตัวแปรที่เก็บข้อมูลเป็นชุดข้อมูล โดยข้อมูลของอาร์เรย์จะเก็บในรูปแบบของลำดับข้อมูล โดยข้อมูลนั้นจะต้องเป็นประเภทเดียวกัน ในการประกาศอาร์เรย์ในภาษา C มันมีรูปแบบคือ

`type name[size];`

โดย type คือประเภทของตัวแปรของอาร์เรย์ที่เราจะสร้าง name เป็นชื่อของตัวแปรอาร์เรย์ และ size เป็นขนาดของอาร์เรย์ การใช้อาร์เรย์ทำให้เราสามารถจัดการข้อมูลได้ง่ายขึ้น กว่าการใช้ตัวแปรธรรมดา

```
int n1 = 10;  
int n2 = 20;  
int n3 = 30;  
int n4 = 40;  
int n5 = 50;
```

Array

เรามีตัวแปร 5 ตัวที่เก็บข้อมูลประเภท integer ไว้ นั่นดูเหมือนจะง่าย เพราะมีแค่ 5 ตัวถ้ามันมีเป็นจำนวนมาก การใช้อาเรย์จะช่วยให้เราสะดวกและแก้ปัญหานี้ได้ แต่ถ้าข้อมูลเยอะมากๆ ล้วนเขียนใหญ่หรือเปล่า ถ้านำมาสร้างเป็นตัวแปรอาเรย์จะได้แบบนี้

```
int n[5] = {10, 20, 30, 40, 50};
```

Declaring an array

การประกาศอาเรย์นั้นคล้ายกับการประกาศตัวแปร เพราะว่าอาเรย์ก็เป็นตัวแปรประเภทหนึ่ง ดังนั้นในการที่จะใช้งานอาเรย์ เราจำเป็นที่จะต้องประกาศมันก่อน

```
int n[5];  
char name[20];  
double m[10];
```



Array

ต่อไปเราจะประยุกต์และกำหนดค่าให้กับอาเรย์ไปพร้อมกัน

```
int n[5] = {5, 10, 20, 30, 50};  
char name[] = {'a', 'e', 'i', 'o', 'u'};  
double m[10];  
m[0] = 1.5;  
m[1] = 4.3;  
m[9] = 50.32;
```

ตัวแปร name เราสามารถกำหนดค่าให้กับตัวแปรอาเรย์ได้ โดยไม่ต้องบอก size ของอาเรย์ โดยภาษา C จะคำนวณให้เราอัตโนมัติ ส่วนตัวแปร m เราได้กำหนดค่าให้กับตัวแปรโดยการใช้อาร์ย์ index

Reading values in an Array

เราจะใช้ index 0 ถึง 7 ในการเข้าถึงค่าตามลำดับ โดย 0 เป็นอ้างอิงตัวแรก

```
#include <stdio.h>

int main()
{
    int n[8] = {5, 10, 15, 20, 25, 30, 35, 40};

    printf("n[2] = %d\n", n[2]);
    printf("n[3] = %d\n", n[3]);

    n[2] = 99;
    printf("n[2] = %d\n", n[2]);
    return 0;
}
```

result:

```
n[2] = 15
n[3] = 20
n[2] = 99
```

Using a for loop with an Array

อาเรย์เป็นลำดับของตัวแปรและใช้ index ในการเข้าถึงข้อมูลมันจึงได้รับความนิยมในการใช้กับ for loop เพื่อที่จะเข้าถึงค่าในอาเรย์

```
#include <stdio.h>

int main()
{
    int n[8] = {5, 10, 15, 20, 25, 30, 35, 40};

    int i;
    for (i = 0; i < 8; i++)
    {
        printf("%d, ", n[i]);
    }

    printf("\n");

    for (i = 7; i >= 0; i--)
    {
        printf("%d, ", n[i]);
    }
    return 0;
}
```

result:

```
5, 10, 15, 20, 25, 30, 35, 40,
40, 35, 30, 25, 20, 15, 10, 5,
```

2D array

อาร์เรย์ 2 มิติ มี row, column ให้จินตนาการว่ามันคือตาราง ซึ่งก็คืออาร์เรย์ของอาร์เรย์ 1 มิติในอง

```
int a[3][4] = {  
    {35, 6, 1},  
    {4, 5, 16},  
    {9, 7, 92}  
};  
  
char b[5][10];
```

```
int i, j;  
for (i = 0; i < 3; i++)  
{  
    for (j = 0; j < 3; j++)  
    {  
        printf("%d, ", a[i][j]);  
    }  
    printf("\n");  
}
```

result:

```
35, 6, 1,  
4, 5, 16,  
9, 7, 92,
```

13.15 see you

A close-up photograph of a healthy meal in a white bowl. The meal consists of a wrap filled with various ingredients, a scoop of brown rice or quinoa, and a variety of fresh microgreens and leafy vegetables. The bowl is set against a dark, blurred background.

lunch

Junior Engineering

Function

- **What is Function in C ?**

ฟังก์ชัน คือ กลุ่มของคำสั่งที่ทำงานเฉพาะอย่าง ช่วยให้โปรแกรม
อ่านง่าย แก้ไขง่าย และนำกลับมาใช้ซ้ำได้
ตัวอย่างที่เคยใช้มาแล้ว: main()

Function

- Why Use Functions

✗ ไม่ใช้ฟังก์ชัน (โค้ดยาว, ซ้ำซ้อน)

```
printf("Hello!\n");
printf("Hello!\n");
printf("Hello!\n");
```

✓ ใช้ฟังก์ชัน (เขียนล้ำ ใช้ชาี่ได้)

```
void sayHello() {
    printf("Hello!\n");
}

int main() {
    sayHello();
    sayHello();
    sayHello();
}
```

Function

- **Function Structure in C**

```
return_type function_name(parameter_list) {  
    // code block  
    return value;  
}
```

ตัวอย่าง

```
int add(int a, int b) {  
    return a + b;  
}
```

- **return_type** (ชนิดของค่าที่ส่งกลับ) เช่น **int, float, char, void**
- **function_name** (ชื่อฟังก์ชัน)
- **parameter_list** (รายการพารามิเตอร์) คือค่าที่ส่งเข้ามาให้ฟังก์ชันใช้งาน ซึ่งอาจมีหลายตัว หรือไม่มีเลยก็ได้
- **return value;** (คำสั่งส่งค่ากลับ) ใช้ส่งค่ากลับออกจากฟังก์ชันไปยังจุดที่เรียกฟังก์ชัน

Function

- **Function Patterns in C**

Pattern	Returns	Takes Arguments	Example
1	No	No	<code>void greet()</code>
2	No	Yes	<code>void greet(char name[])</code>
3	Yes	No	<code>int getNumber()</code>
4	Yes	Yes	<code>int add(int a, int b)</code>

Function

● Variable Scope in Functions

- Local Variables (ตัวแปรภายในฟังก์ชัน): ประกาศและใช้
งานได้เฉพาะภายในฟังก์ชันนั้นเท่านั้น
- Global Variables (ตัวแปรภายนอกฟังก์ชัน): ประกาศนอก
ฟังก์ชันทั้งหมด สามารถใช้ได้ทุกฟังก์ชันในไฟล์เดียวกัน
- Static Variables (ตัวแปรคงค่า): ประกาศในฟังก์ชัน แต่
ค่าจะไม่หายไปเมื่อฟังก์ชันจบการทำงาน ใช้ได้เฉพาะใน
ฟังก์ชันนั้น

```
int x = 5; // Global

void show() {
    int y = 10; // Local
    static int count = 0; // Static
}
```

Function

ผลลัพธ์ของการ run program นี้เป็นอย่างไร

```
#include <stdio.h>

int y = 100; // Global variable

void test() { // Function Definition
    int y = 10; // Local variable
    printf("Inside function: y = %d\n", y);
}

int main() {
    test(); // Function Call
    printf("In main: y = %d\n", y);
    return 0;
}
```

→ A

Inside function: y = 10
In main: y = 100

B

Inside function: y = 100
In main: y = 100

C

Inside function: y = 10
In main: y = 10

Function

- **Function Declaration**

✗ Incorrect Example (Function Defined After main() Without Declaration)

```
#include <stdio.h>

int main() {
    int result = add(3, 4); // ✗ Error: Compiler doesn't know what 'add' is
    printf("Result: %d\n", result);
}

int add(int a, int b) {
    return a + b;
}
```

Function

- **Function Declaration**

-  **Corrected Example Using Function Declaration**

```
#include <stdio.h>

// Function declaration (prototype)
int add(int, int);

int main() {
    int result = add(3, 4); // ✅ No error – compiler knows about 'add'
    printf("Result: %d\n", result);
}

// Function definition
int add(int a, int b) {
    return a + b;
}
```

Function

- **Function Declaration**

Example: Using a Function Across Multiple Files

1. mathlib.h – Header File (Function Declaration):

```
// mathlib.h
int add(int a, int b);
```

2. mathlib.c – Source File (Function Definition):

```
// mathlib.c
#include "mathlib.h"

int add(int a, int b) {
    return a + b;
}
```

Function

- **Function Declaration**

Example: Using a Function Across Multiple Files

3. main.c – Main Program File:

```
// main.c
#include <stdio.h>
#include "mathlib.h"

int main() {
    int result = add(10, 15);
    printf("10 + 15 = %d\n", result);
    return 0;
}
```

Function

- **Function Declaration**

-  **Native Function Usage in C (Without Prototypes)**

```
#include <stdio.h>

int add(int a, int b) { // Function Definition
    return a + b;
}

int main() {
    int result = add(3, 4); // Function Call
    printf("Result: %d\n", result);
}
```

Function

• Complete Example

```
#include <stdio.h>

// Function Declarations (Prototypes)
void hello();                                // Pattern 1
void printName(char name[]);                  // Pattern 2
int getNumber();                               // Pattern 3
int sum(int a, int b);                        // Pattern 4

int main() {
    // Pattern 1: No return, no parameter
    hello();

    // Pattern 2: No return, with parameter
    printName("Alice");

    // Pattern 3: Return value, no parameter
    int number = getNumber();
    printf("getNumber() returned: %d\n", number);

    // Pattern 4: Return value, with parameters
    int result = sum(5, 7);
    printf("sum(5, 7) returned: %d\n", result);

    return 0;
}
```

```
// Pattern 1: No return, no parameter
void hello() {
    printf("Hello from function with no parameters and no return!\n");
}

// Pattern 2: No return, with parameter
void printName(char name[]) {
    printf("Hello, %s!\n", name);
}

// Pattern 3: Return value, no parameter
int getNumber() {
    return 42;
}

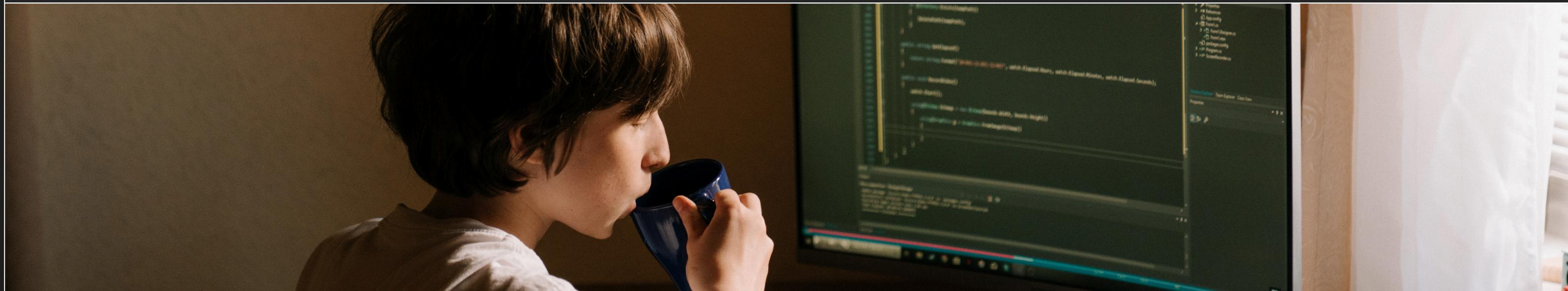
// Pattern 4: Return value, with parameters
int sum(int a, int b) {
    return a + b;
}
```



Search

Let's Take a Break 5 Minutes!

Breakslides



CODEBOTIX

Pointer

- **What is a Pointer?**

ตัวแปรที่เก็บ "ที่อยู่ของหน่วยความจำ" ของตัวแปรอื่น

Pointer

• Pointer & Memory

ทุกตัวแปรในภาษาซี ประกอบด้วย:

- variable name (ชื่อตัวแปร)
- value (ค่าที่ตัวแปรนั้นเก็บอยู่)
- address (ที่อยู่ของตัวแปรนั้นใน memory)

Pointer

• Pointer & Memory

```
int x, y;  
  
int *a;  
int *b;  
  
x = 10;  
a = &x;  
b = a;  
y = *b;  
  
printf("address of x: %p\n", &x);  
printf("value of x: %d\n\n", x);  
printf("address of a: %p\n", &a);  
printf("value of a: %p\n", a);  
printf("dereference of a: %d\n\n", *a);  
printf("address of b: %p\n", &b);  
printf("value of b: %p\n", b);  
printf("dereference of b: %d\n\n", *b);  
printf("address of y: %p\n", &y);  
printf("value of y: %d\n", y);
```

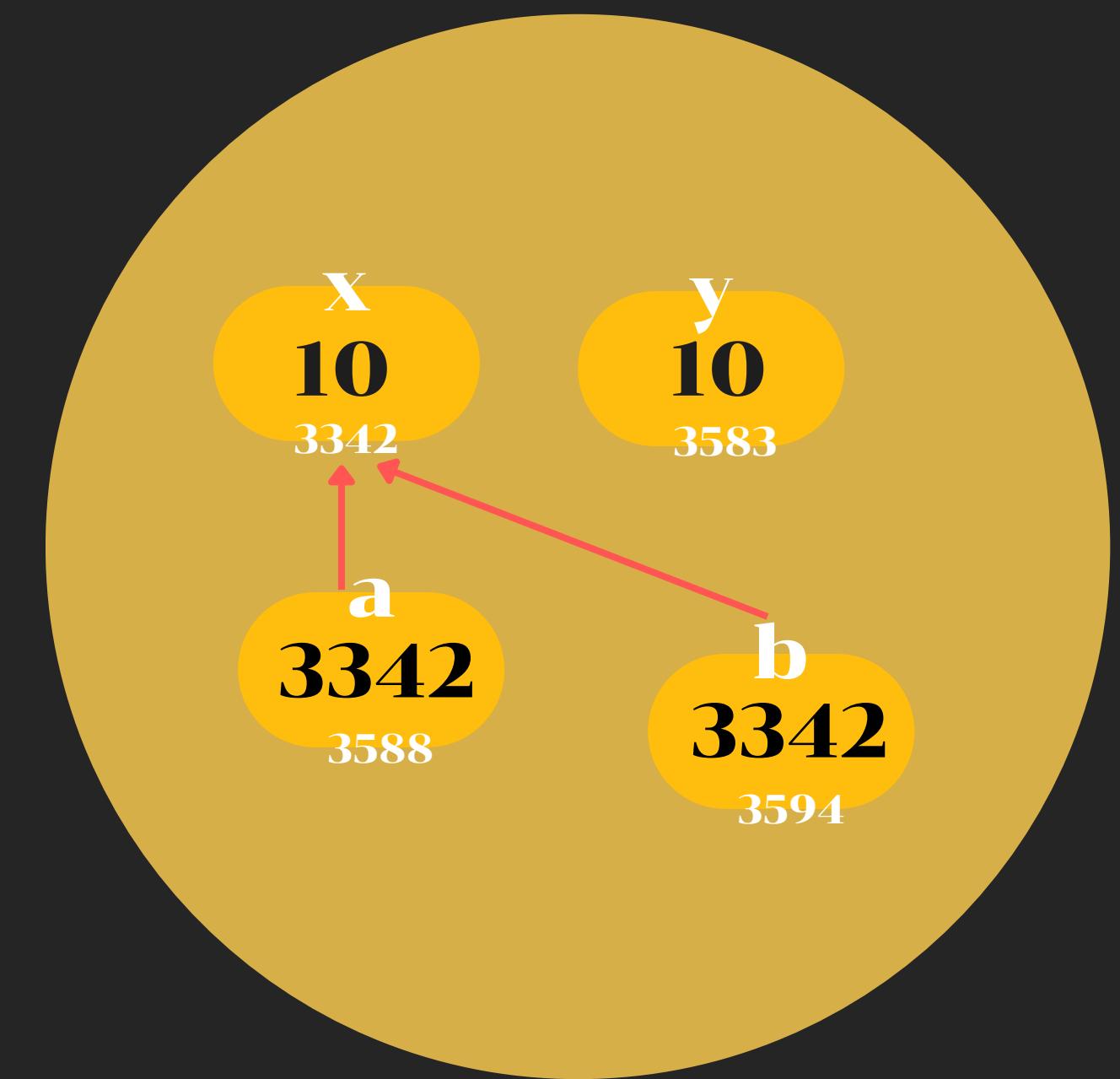
address of x: 0x7ffd6d07576c
value of x: 10

address of a: 0x7ffd6d075760
value of a: 0x7ffd6d07576c
dereference of a: 10

address of b: 0x7ffd6d075758
value of b: 0x7ffd6d07576c
dereference of a: 10

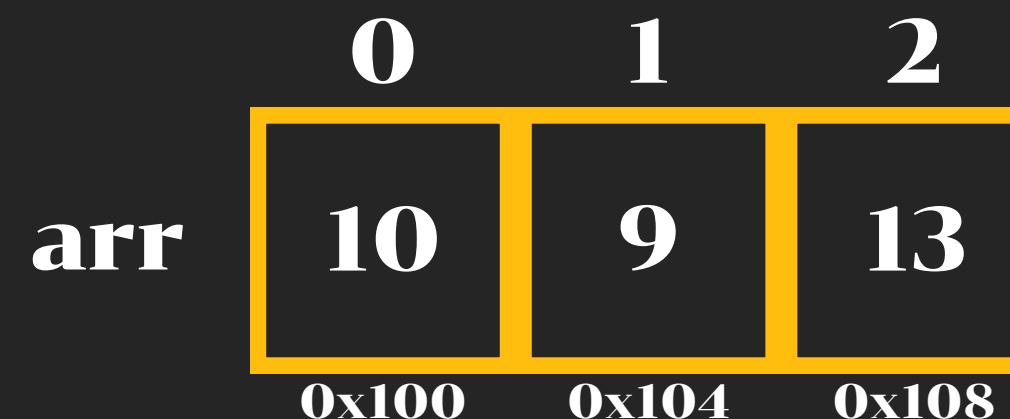
address of y: 0x7ffd6d075768
value of y: 10

Computer Memory



Pointer

- **Pointer and Array**



```
int arr[3] = {10, 9, 13};

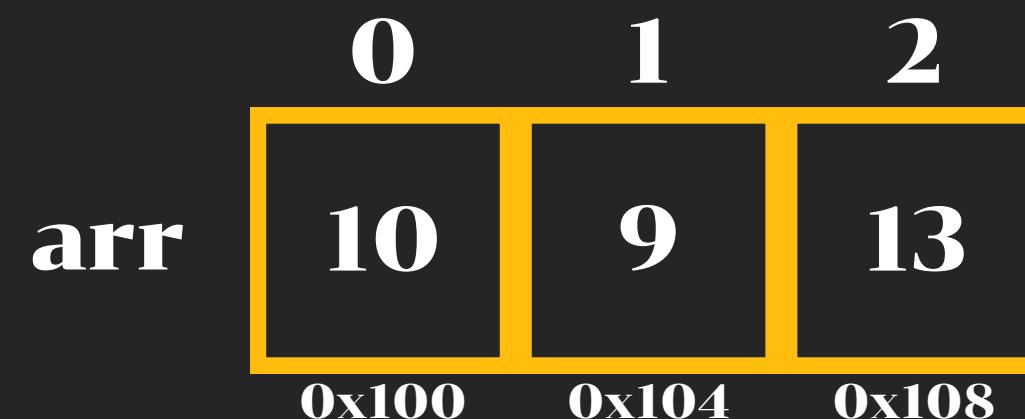
printf("%p\n", arr); // same as &arr[0]
printf("%p\n", &arr[0]);
printf("%p\n", &arr[1]);
printf("%p\n", &arr[2]);

printf("%d\n", *arr); // same as arr[0]
printf("%d\n", *(arr+1)); // same as arr[1];
```

```
0x7ffcacfa3a04
0x7ffcacfa3a04
0x7ffcacfa3a08
0x7ffcacfa3a0c
10
9
```

Pointer

- Pointers and Arrays



```
int arr[3] = {10, 9, 13};

printf("%p\n", arr); // same as &arr[0]
printf("%p\n", &arr[0]);
printf("%p\n", &arr[1]);
printf("%p\n", &arr[2]);

printf("%d\n", *arr); // same as arr[0]
printf("%d\n", *(arr+1)); // same as arr[1];
```

```
0x7ffcacfa3a04
0x7ffcacfa3a04
0x7ffcacfa3a08
0x7ffcacfa3a0c
10
9
```

Pointer

- **Function: Passing by Value vs Reference**

ในภาษา C มี 2 วิธีในการส่งค่าผ่านฟังก์ชัน:

1. Pass by Value (ส่งค่าด้วยตัวแปร)

- ฟังก์ชันจะได้รับ สำเนา (copy) ของค่าตัวแปร
- การเปลี่ยนแปลงภายในฟังก์ชัน ไม่ส่งผลต่อค่าตัวแปรเดิม

2. Pass by Reference (ส่งค่าด้วยการอ้างอิง)

- ฟังก์ชันจะได้รับ ตำแหน่งหน่วยความจำ (address) ของตัวแปร
- ใช้ pointer เพื่อเข้าถึงและแก้ไขค่าตัวแปรต้นฉบับ
- การเปลี่ยนแปลงจะ ส่งผลต่อค่าจริง

Pointer

- **Function: Passing by Value vs Reference**

```
#include <stdio.h>

// Pass by Value
void modifyValue(int a) {
    a = a + 10;
    printf("Inside modifyValue: a = %d\n", a);
}

// Pass by Reference
void modifyReference(int *a) {
    *a = *a + 10;
    printf("Inside modifyReference: a = %d\n", *a);
}

int main() {
    int num1 = 5, num2 = 5;

    printf("Before modifyValue: num1 = %d\n", num1);
    modifyValue(num1);
    printf("After modifyValue: num1 = %d\n\n", num1);

    printf("Before modifyReference: num2 = %d\n", num2);
    modifyReference(&num2);
    printf("After modifyReference: num2 = %d\n", num2);

    return 0;
}
```

Before modifyValue: num1 = 5
Inside modifyValue: a = 15
After modifyValue: num1 = 5

Before modifyReference: num2 = 5
Inside modifyReference: a = 15
After modifyReference: num2 = 15



Search

Let's Take a Break 5 Minutes!

Breakslides



CODEBOTIX

Struct

- **What is a Struct in C**

Struct គឺជាប្រភេទប៊ូមូលកំណត់ដោយខ្លួន (user-defined data type) ដែលអាចរាយការណ៍បច្ចុប្បន្ន ឬបង្កើតឡាយការងារ សម្រាប់ប្រព័ន្ធឌីជីថល ឬកិច្ចការណ៍។ ក្នុង Struct នឹងរាយការណ៍បច្ចុប្បន្ន ឬបង្កើតឡាយការងារ ដូចជា ឈ្មោះ ឈ្មោះ ឬអាសយដ្ឋាន។

Struct

- **Structure of a struct in C**

```
struct StructName {  
    data_type member1;  
    data_type member2;  
    ...  
};
```

- StructName: ថែបងក្រសាន្ត ជូន Student, Employee
- data_type: ប្រព័ន្ធដែលបងសមាជិក ជូន int, float, char[]
- member1, member2: ថែបងសមាជិក (field) កាយត្រួតសរុច

```
struct Student {  
    int id;  
    char name[50];  
    float score;  
};
```

Struct

- Example of struct

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

struct Student {
    int id;
    char name[50];
    float score;
};
```

```
int main() {
    struct Student s1;

    s1.id = 1;
    strcpy(s1.name, "สมชาย");
    s1.score = 89.5;

    printf("id: %d\n", s1.id);
    printf("name: %s\n", s1.name);
    printf("score: %.2f\n", s1.score);

    return 0;
}
```

Struct

- **Example of struct**

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

typedef struct {
    int id;
    char name[50];
    float score;
} Student;
```

```
int main() {
    Student s1;

    s1.id = 101;
    strcpy(s1.name, "สมชาย");
    s1.score = 88.5;

    Student *ptr = &s1;

    printf("id: %d\n", ptr->id); // same as (*ptr).id
    printf("name: %s\n", ptr->name); // same as (*ptr).name
    printf("score: %.2f\n", ptr->score); // same as (*ptr).score

    return 0;
}
```

Struct

- Example of struct

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

typedef struct {
    int id;
    char name[50];
    float score;
} Student;

void printStudents(Student arr[], int size) {
    for (int i = 0; i < size; i++) {
        printf("Student %d:\n", i + 1);
        printf("  ID: %d\n", arr[i].id);
        printf("  Name: %s\n", arr[i].name);
        printf("  Score: %.2f\n\n", arr[i].score);
    }
}
```

```
int main() {
    Student student_list[] = {
        {101, "Somchai", 88.5},
        {102, "Suda", 91.0},
        {103, "Anan", 76.3}
    };

    int size = sizeof(student_list) / sizeof(student_list[0]);

    printStudents(student_list, size);

    return 0;
}
```

Student 1:
ID: 101
Name: Somchai
Score: 88.50

Student 2:
ID: 102
Name: Suda
Score: 91.00

Student 3:
ID: 103
Name: Anan
Score: 76.30



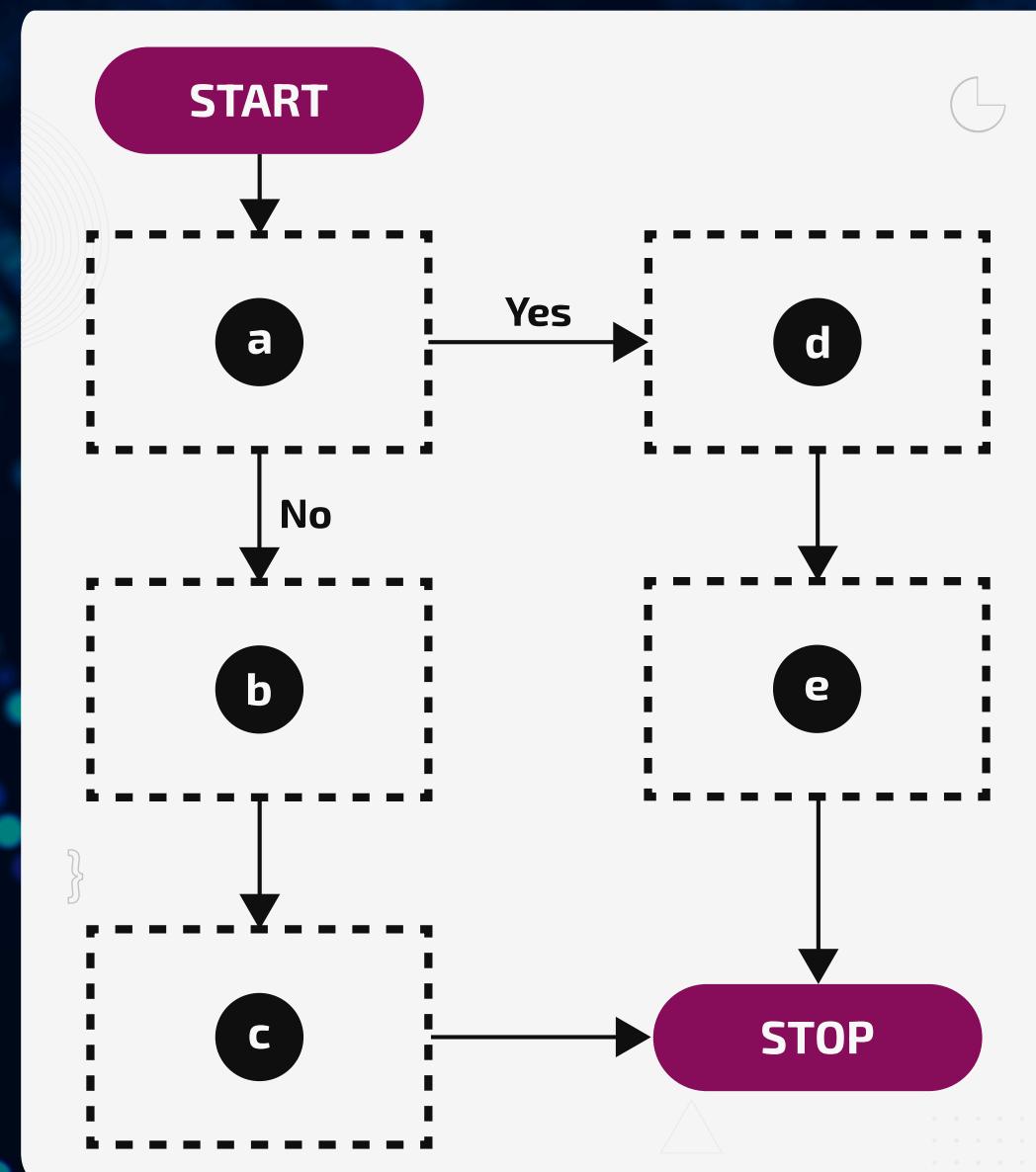
Search

Let's Take a Break 5 Minutes!

Breakslides



CODEBOTIX



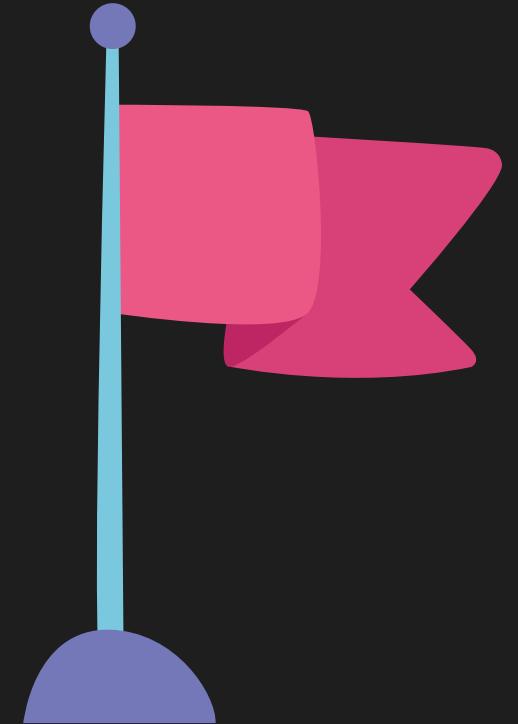
FLOWCHART

การแสดงขั้นตอนวิธีด้วยผังงาน หรือ รหัสเกี่ยม การเตรียมออกแบบโค้ดล่วงหน้า

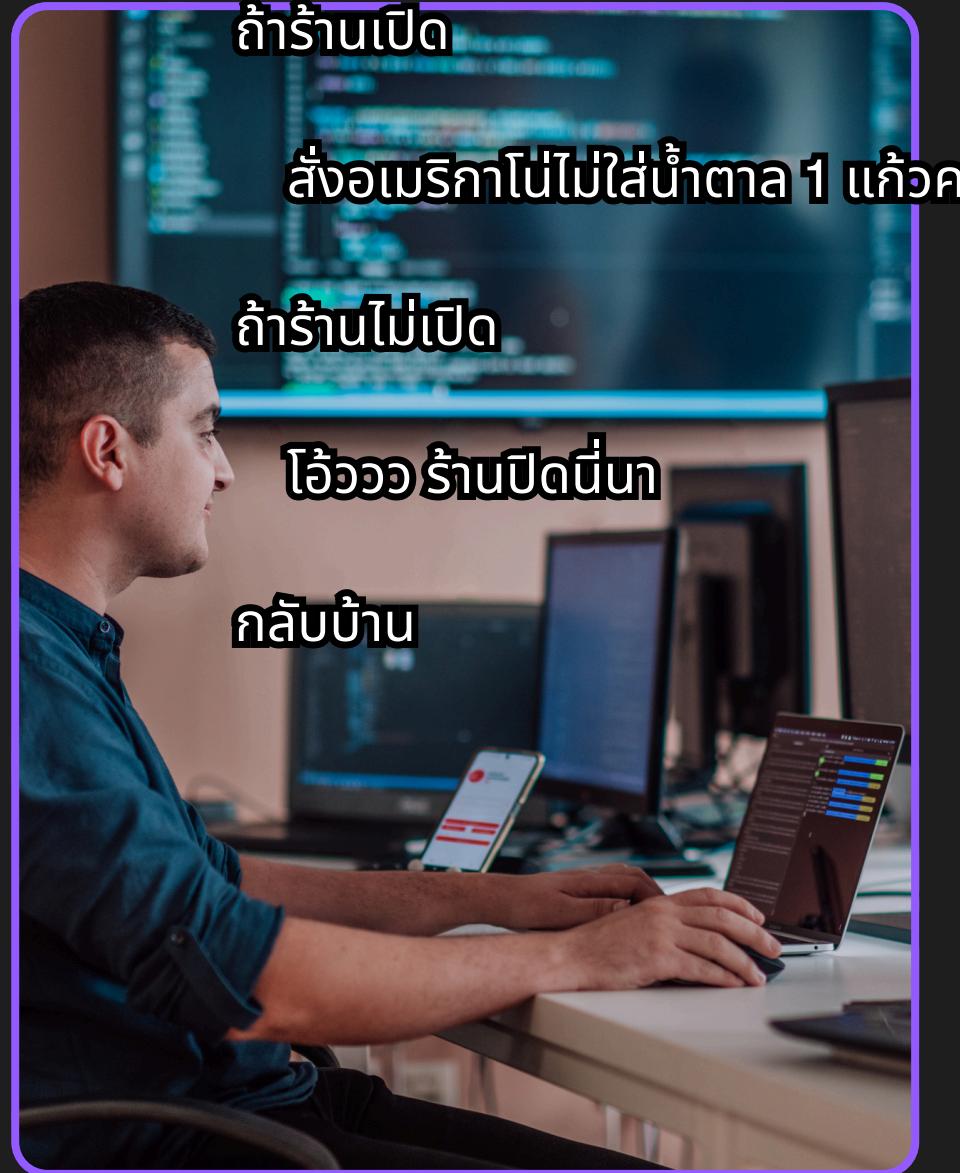
pseudo code

คือ โค้ดง่าย ๆ กี่เราไม่ต้องเขียน
โปรแกรม

แต่เราคิดกระบวนการที่ไปดำเนิน
ผลลัพธ์ โดยที่มีเงื่อนไขเป็นตัวแยก
อัลกอริทึมหรือไม่มีก็ได้



เดินไปซื้อกาแฟ



```
def buy_coffee():
    if cafe_open:
        print("สั่งอเมริกาโน่ไม่ใส่น้ำตาล 1 แก้วครับ")
    else:
        print("哦อวว ร้านปิดนี่นา")
    print("กลับบ้าน")
```

buy_coffee()



Common flowchart symbols overview

ສัญลักษณ์เหล่านี้ใช้ในการสร้างผังงานและแสดงวิธีแก้ไขปัญหาแบบ Software Flow Design กีລະขันตอน

Symbol	Name	Usage
	Line	Represents the flow from one component to the next
	Process	An action
	Input/Output	An input or output
	Decision	A yes/no/true/false decision
	Terminal	The start or end of the process

Common flowchart symbols meaning

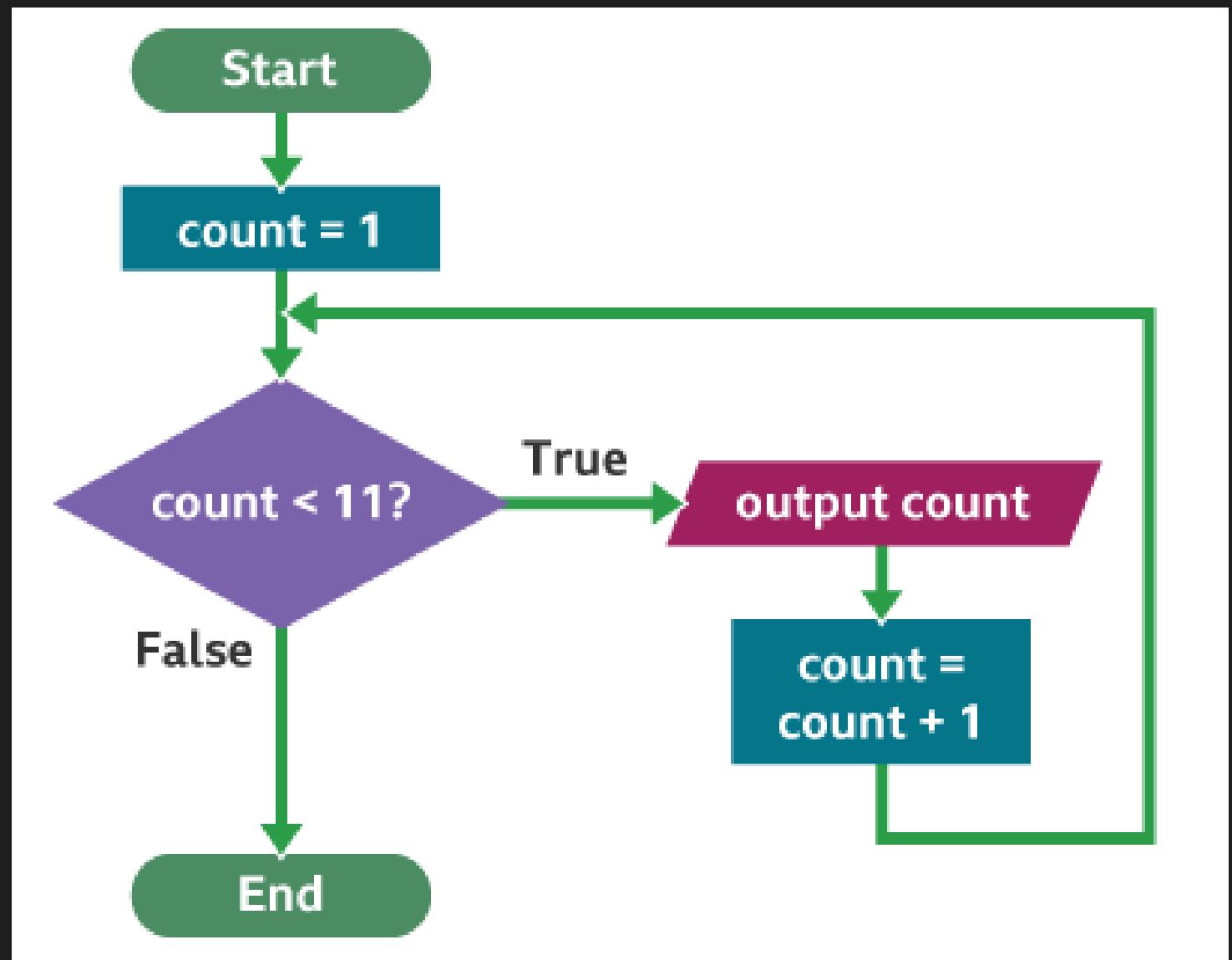
นอกจากนี้ถ้าเราแบ่งตามกระบวนการใช้งานจะแบ่งได้ดังนี้

Terminator	
Direction	
Process	
Input/Output	
Decision	
Connector	
Offpage Connector	

Annotation สำหรับใส่คอมเมนต์ที่ไม่มีส่วนเกี่ยวข้องกับลำดับวิธีโดยตรง	
Predefined Process / Module สำหรับเรียกโปรแกรมย่อยที่เตรียมไว้แล้ว	
Preparation สำหรับเตรียมตัวแปร ในการทำงานวนซ้ำแบบ for loop	

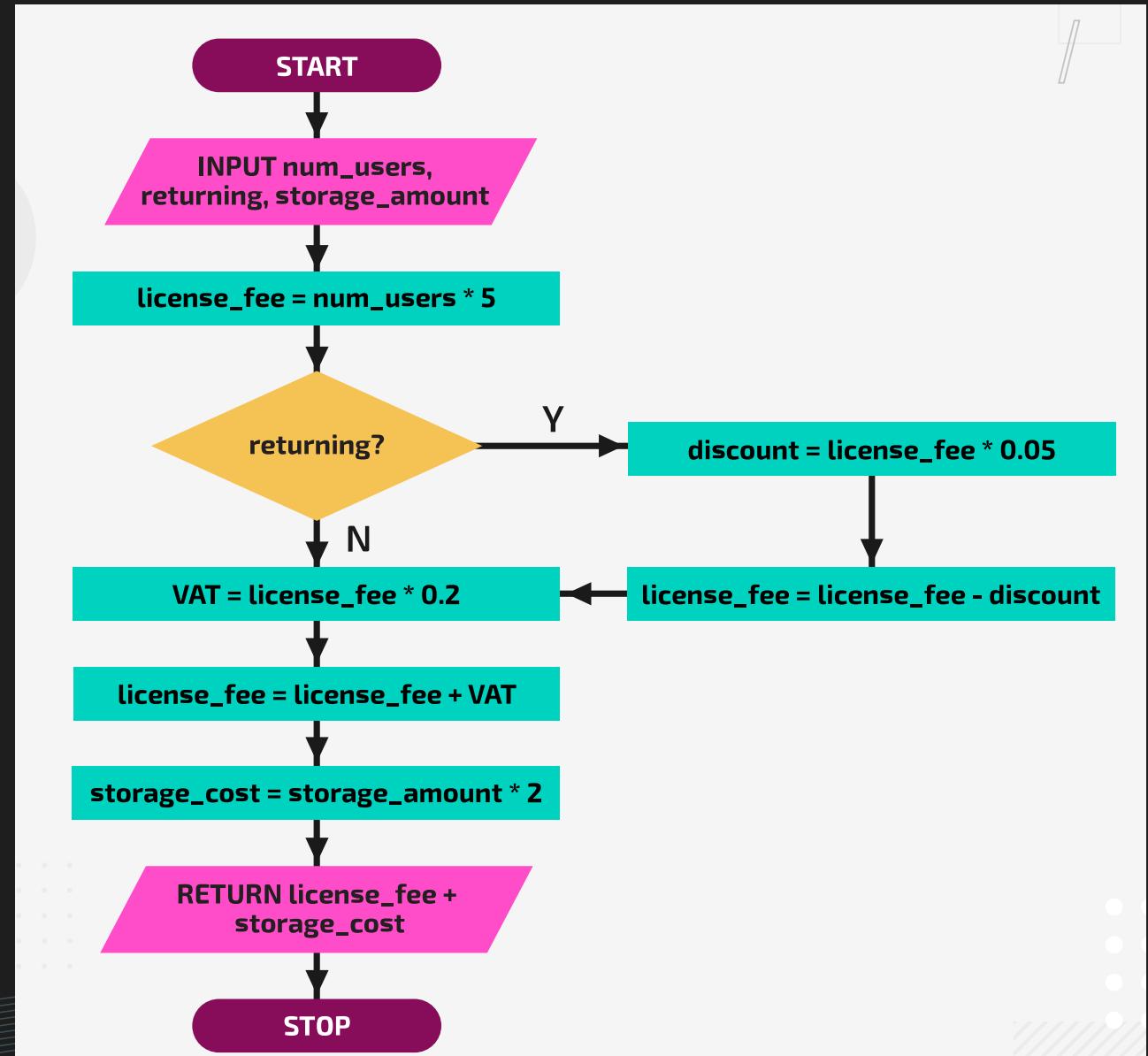
example 1

ผังงาน (Flowchart) สามารถใช้เพื่อวางแผนโปรแกรมได้ ผังงานง่ายๆ นี้จะแสดงอัลกอริทึมสำหรับโปรแกรมที่พิมพ์ตัวเลข 1 ถึง 10



example 2

เขียนโปรแกรมคำนวณรับค่าจำนวนคน, การมีส่วนลด, เงินค่าซื้อ software product แต่ละปี จากนั้นจะเริ่มคิดค่าธรรมเนียม License โดยคิดจากจำนวนคน คนละ 5 บาท และถ้ามีส่วนลด จะลดให้ 5% แต่ถ้าไม่มีจะคิด VAT กับที่ มีส่วนลดก็ต้องเสียภาษี (ภาษีมูลค่าเพิ่ม) 20% ของ ค่าธรรมเนียม จากนั้นต้องจ่ายค่าซื้อ software product 2 เท่าของกับค่าธรรมเนียมและภาษีเพื่อซื้อดาว



Problem solve

โจทย์ที่ 1 การหาค่าสัมบูรณ์ (Absolute Value) คือ ค่าที่ไม่ว่ายังไงก็จะเป็น正值แม้จะใส่ลบเข้าไป
เขียน flowchart สำหรับหาค่าสัมบูรณ์ของตัวเลขที่ผู้ใช้ป้อน

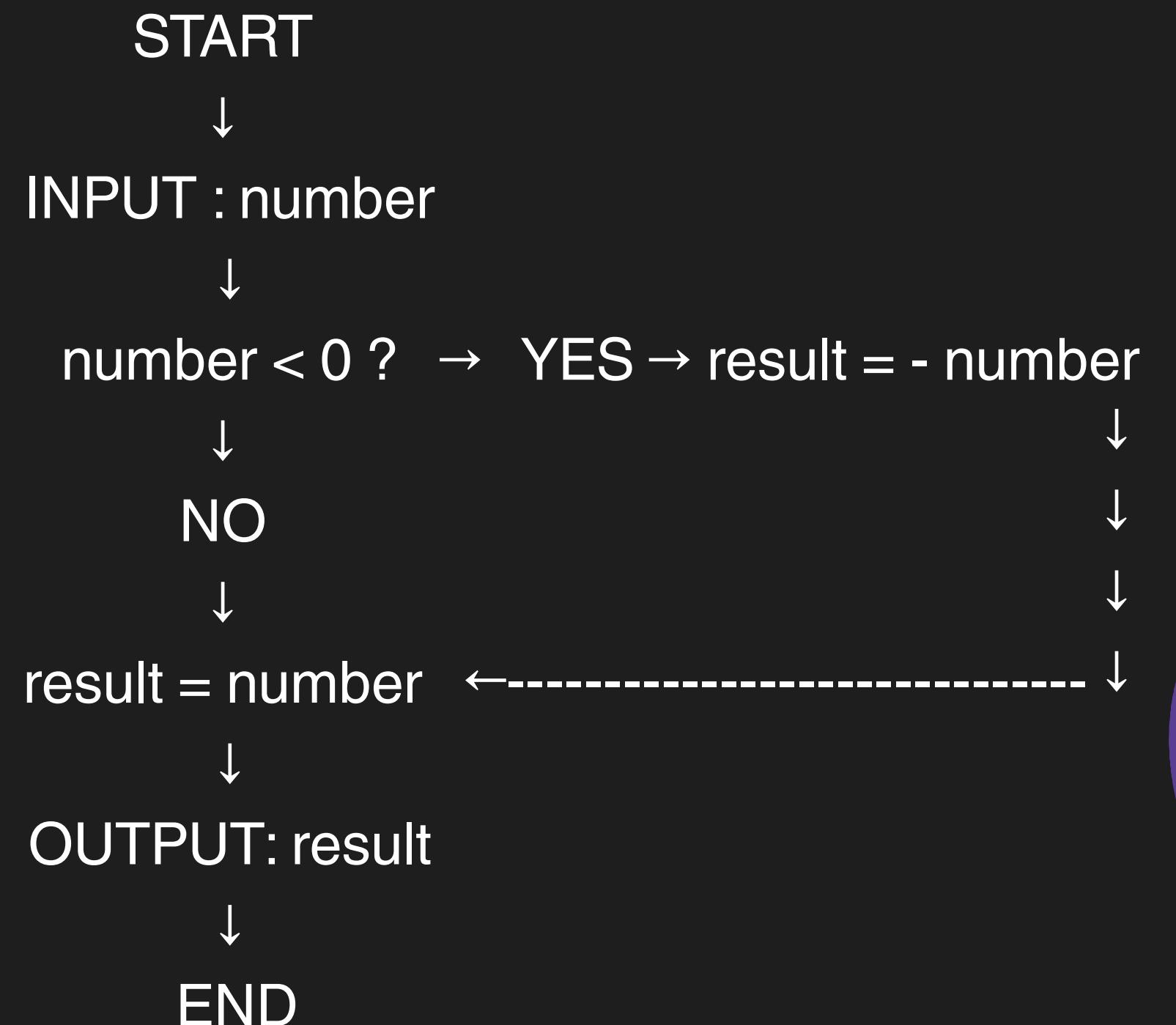
Problem solve

โจทย์ที่ 1 การหาค่าสัมบูรณ์ (Absolute Value) คือ ค่าที่ไม่ว่ายังไงก็จะเป็น正值เสมอแม้จะใส่ลบเข้าไป
เขียน flowchart สำหรับหาค่าสัมบูรณ์ของตัวเลขที่ผู้ใช้ป้อน

EASY

Answer

ໂຈກຍົກ 1



Problem solve

โจทย์ที่ 2 การหาค่าสูงสุดจาก 3 ตัวเลข

เขียน flowchart สำหรับหาค่าสูงสุดจากตัวเลข 3 ตัวที่ผู้ใช้ป้อน

MEDIUM

Problem solve

โจทย์ที่ 3 การหาจำนวนเฉพาะ (Prime Number)
เขียน flowchart เพื่อตรวจสอบว่าตัวเลขที่ป้อนเป็นจำนวนเฉพาะหรือไม่

HARD

Problem solve

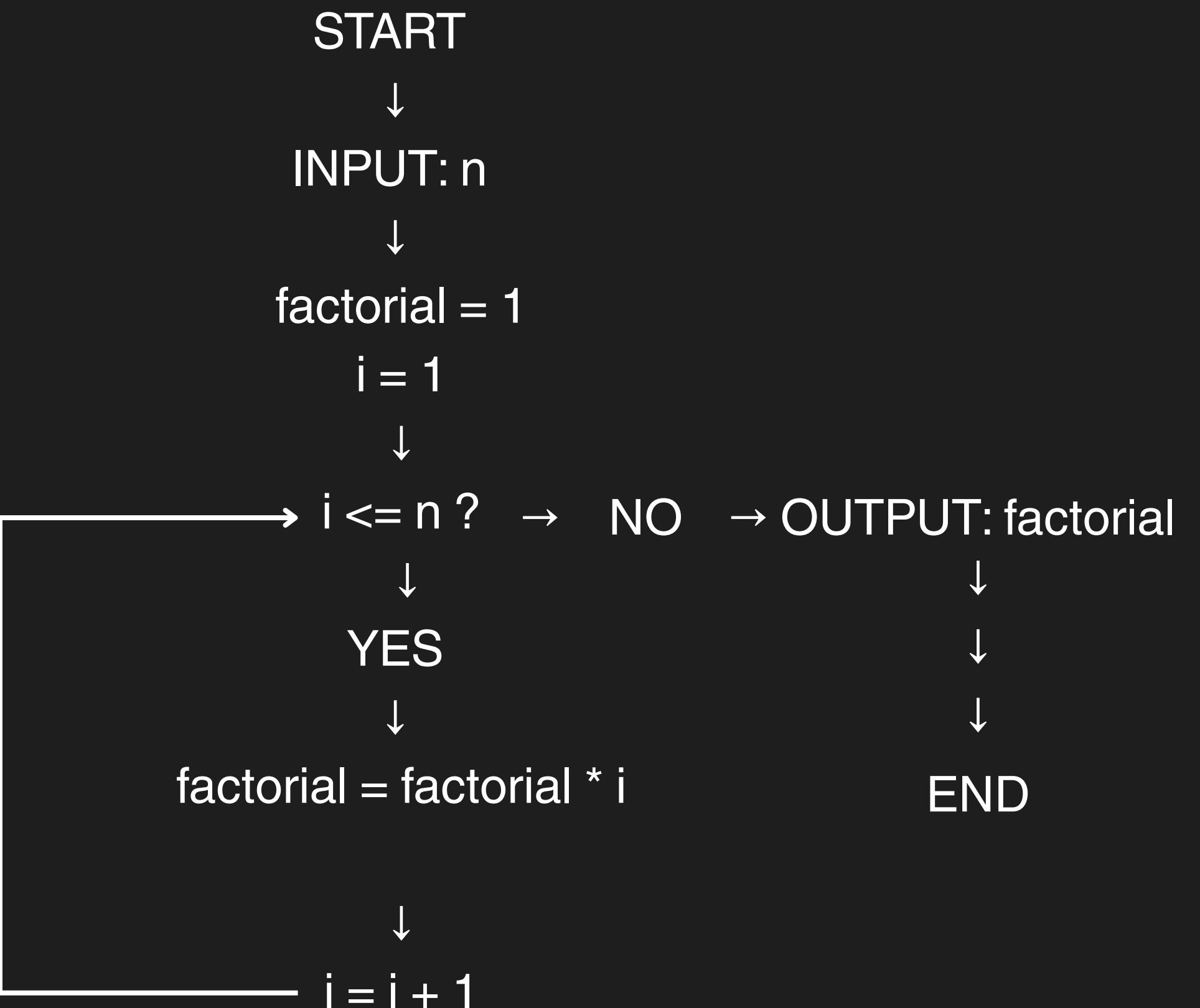
โจทย์ที่ 3 การคำนวนแฟกторเรียล

เขียน flowchart สำหรับคำนวนค่าแฟกตอเรียลของตัวเลข n

HARD

Answer

ໂຈກຍົກໍ 4



Challenges on Practice questions

ต่อไปจะให้บอท ๆ สุ่มจับมาคุยกับห้อง
Channel discord เพื่อฝึกฝน



P' Blue

programming.in.th

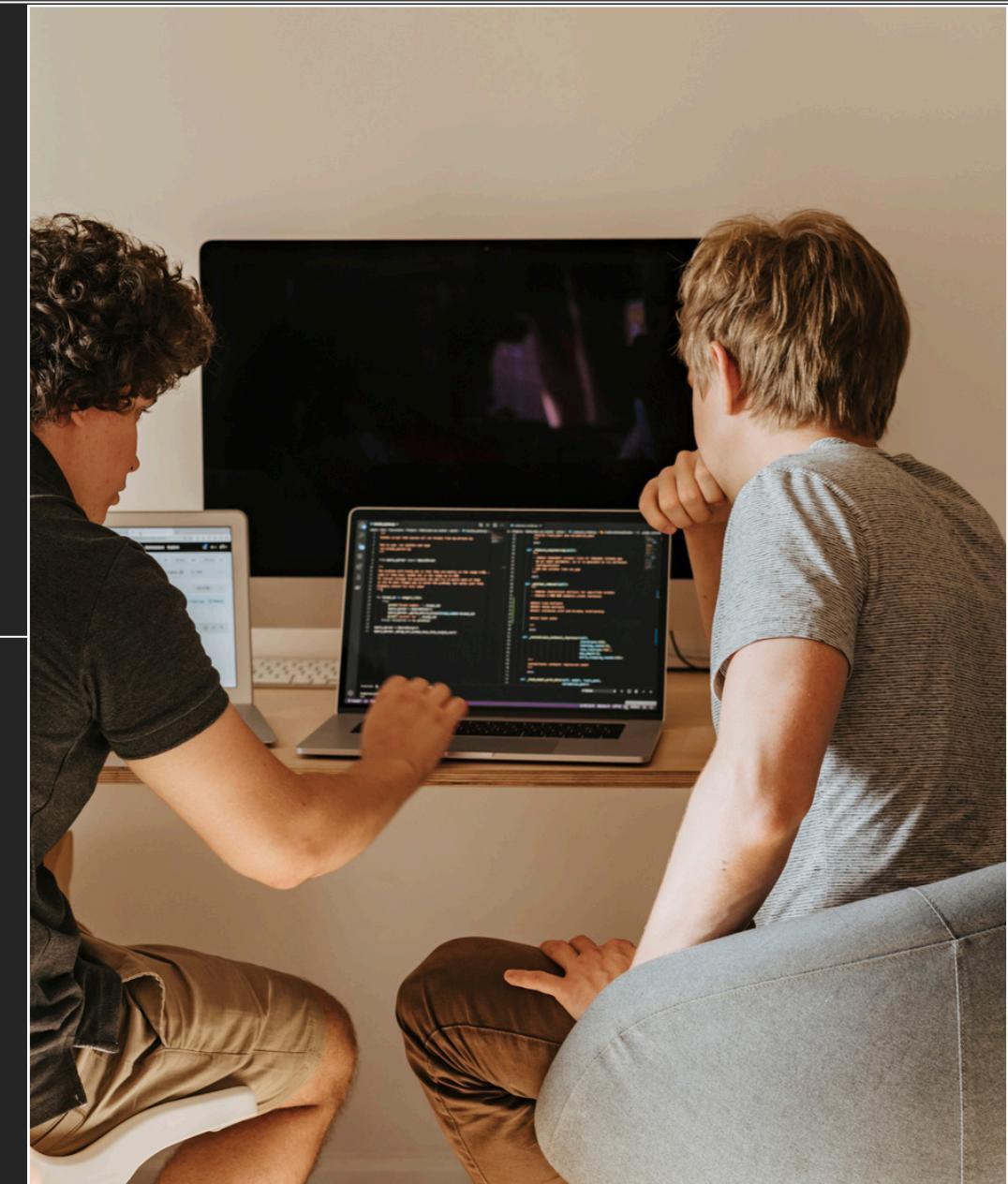
ทั้งหมด วิดีโอ รูปภาพ ข้อค้นห์ วิดีโอด้วย ข่าวสาร หนังสือ เพิ่มเติม ▾ เครื่องมือ ▾

programming.in.th
https://programming.in.th ::

ทำไมต้อง Programming.in.th

programming.in.th เป็นแหล่งรวมโจทย์ competitive programming ที่มีความหลากหลายมากที่สุดในไทยแล้วครับ สำหรับใครที่กำลังตามหาโจทย์ในการเรียนฝึกทักษะการเขียนโปรแกรมของตัวเองอยู่ ผู้คนแนะนำ ...

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus porta pellentesque purus non luctus. Maecenas pellentesque bibendum orci quis vulputate. Aliquam blandit auctor accumsan. Cras dictum arcu tortor, eget imperdiet dui pharetra volutpat. Donec ornare, metus non rutrum maximus, tortor leo posuere mi, vel iaculis mauris orci ut ligula.



Join Coding Communities

Practice Regularly

Contribute to The Community

P' Blue

programming.in.th

Home Tasks Archive Learn About

Profile picture

Tasks

Browse Over 800+ Tasks

Search...

All	Problem Title	<input type="checkbox"/> Show tag	Solved	Score
Tried	A+B Problem 0000	Show all tag >	11244	0 points
Solved	Grading 0001	Show all tag >	7472	0 points
Bookmarked	Min Max 0002	Show all tag >	6081	0 points
	Matrix Addition 0003	Show all tag >	3886	0 points
	Character Checker 0004	Show all tag >	3969	0 points
	Pythagorus 0005	Show all tag >	3789	0 points

All

Tried

Solved

Bookmarked

Tag Filter clear

- Implementation
- Math
- Ad-hoc
- String
- Data structure
- DP
- Sweepline
- Greedy
- DFS
- Bitmasks
- Binary search
- Graph

Star icon

Problem 1

Min Max

1 second, 64 megabytes

โจทย์ จงหาค่าน้อยที่สุด และค่ามากที่สุด จากข้อมูลที่กำหนดให้ และแสดงผลออกทางจอภาพ

ข้อมูลนำเข้า

บรรทัดแรก จำนวนเต็มบวก n ($1 \leq n \leq 1\,000$) บ่งบอกถึงจำนวนข้อมูลที่โจทย์กำหนดให้

บรรทัดที่ 2 ถึง $n + 1$ จำนวนเต็ม A_i เป็นข้อมูลทั้งหมด ($-2\,000\,000\,000 \leq A_i \leq 2\,000\,000\,000$)

ข้อมูลส่งออก

บรรทัดแรก จำนวนเต็ม m แสดงจำนวนที่มีค่าน้อยที่สุดในชุดข้อมูลที่โจทย์กำหนด

บรรทัดที่สอง จำนวนเต็ม M แสดงจำนวนที่มีค่ามากที่สุดในชุดข้อมูลที่โจทย์กำหนด

ตัวอย่างข้อมูลนำเข้าและข้อมูลส่งออก

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลส่งออก
5	1
1	5
2	
3	
4	
5	

Problem 2

programming
.in.th

Matrix Addition

1 second, 64 megabytes

โจทย์ บวกเมทริกซ์ 2 เมทริกซ์ มิติ $m \times n$ จากที่โจทย์ระบุให้

ข้อมูลนำเข้า

บรรทัดแรก จำนวนเต็มบวก m, n ($1 \leq m, n \leq 100$) แสดงมิติของเมทริกซ์

บรรทัดที่ 2 ถึงบรรทัดที่ $m + 1$ จำนวนเต็ม n จำนวนในแต่ละบรรทัด แสดงสมาชิกของเมทริกซ์ที่หนึ่ง

บรรทัดที่ $m + 2$ ถึงบรรทัดที่ $2 \cdot m + 1$ จำนวนเต็ม n จำนวนในแต่ละบรรทัด แสดงสมาชิกของเมทริกซ์ที่สอง คุณสามารถสมมติได้ว่าสมาชิกของเมทริกซ์แต่ละตัวนั้น มีค่าไม่ต่ำกว่า $-2\,000\,000\,000$ และไม่เกิน $2\,000\,000\,000$

ข้อมูลส่งออก

จำนวนบรรทัดทั้งสิ้น m บรรทัด แต่ละบรรทัดแสดงสมาชิกของเมทริกซ์ผลลัพธ์ ในรูปแบบตามข้อมูลนำเข้า

ตัวอย่างข้อมูลนำเข้าและข้อมูลส่งออก

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลส่งออก
3 3	2 3 4
1 2 3	4 3 2
3 2 1	2 4 3
1 3 2	
1 1 1	
1 1 1	
1 1 1	

Answer Problem 2

```
#include <stdio.h>

int main() {
    int m, n;

    scanf("%d %d", &m, &n);

    int A[m][n], B[m][n];
```

รับข้อมูล Matrix A

```
for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++) {
        scanf("%d", &A[i][j]);
    }
}
```

รับข้อมูล Matrix B

```
for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++) {
        scanf("%d", &B[i][j]);
    }
}
```

**คำนวณและแสดงผลการบวก
matrix A + B**

```
for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++) {
        printf("%d", A[i][j] + B[i][j]);
        if (j < n - 1) printf(" ");
    }
    printf("\n"); // ขึ้นบรรทัดใหม่
}

return 0;
}
```



Problem solve

โจทย์ใหม่ – “รถเมล์คันสุดท้าย”

ที่ป้ายรถเมล์ในเมือง มีรถเมล์วิ่งผ่านหลายสายต่อวัน
แต่ละสายมี หมายเลขสายรถเมล์ เป็นจำนวนเต็ม

วันนี้คุณมาขึ้นรถที่ป้าย แล้วจดหมายเลขสายรถเมล์ที่วิ่งผ่านทั้งหมด
สิ่งที่คุณอยากรู้คือ สายรถเมล์คันสุดท้ายที่ผ่านมาก็คือสายอะไร

Input

- บรรทัดแรก: จำนวนเต็ม N (จำนวนรถเมล์ที่ผ่าน)
- บรรทัดถัดมา: หมายเลขสายรถเมล์ N ค่า (คั่นด้วย space)

Output

- หมายเลขสายรถเมล์คันสุดท้าย

ตัวอย่าง 1

makefile

Copy Edit

Input:

5
8 11 23 15 42

Output:

42

ตัวอย่าง 2

makefile

Copy Edit

Input:

3
101 99 77

Output:

77

Problem solve

โจทย์ 1 (ง่าย) – “เสียงสะท้อนในถ้ำ”

ในป่าลึก คุณเดินเข้าไปในถ้ำที่มีดสนิท เสียงก้องสะท้อนกลับมาเป็นประโยคภาษาอังกฤษหนึ่งบรรทัด คุณสังเกตว่าเสียงสะท้อนทำให้ได้ยินสระดังขัดเป็นพิเศษ จงเขียนโปรแกรมหาว่า ถ้าฟังจนจบ เสียงสะท้อนนี้มีสระกี่ตัว

Input Example

```
nginx
```

 Copy  Edit

```
Hello Cave
```

Output Example

 Copy  Edit

```
4
```

Problem solve



โจทย์ – “จดหมายลับของราชากษัตริย์”

ราชากษัตริย์ต้องจดหมายลับ แต่ทุกตัวอักษรกลับพิมพ์ผิด
คุณในฐานะผู้ช่วยต้องแก้ไขจดหมาย โดย สลับพิมพ์ตัวอักษรทุกตัว:

- ตัวพิมพ์เล็ก → พิมพ์ใหญ่
- ตัวพิมพ์ใหญ่ → พิมพ์เล็ก
- ตัวเลขและสัญลักษณ์ไม่เปลี่ยน

จงเขียนโปรแกรมที่ รับข้อความ string และแปลงตัวอักษรตามกฎ

Input

- ข้อความ 1 บรรทัด

Output

- ข้อความที่แปลงตัวอักษรแล้ว

ตัวอย่าง

Example 1

makefile

Copy Edit

Input:

Hello World!

Output:

hELLO wORLD!

Example 2

makefile

Copy Edit

Input:

AbC123xYz

Output:

aBc123XyZ

Problem solve

โจทย์ 3 (ยาก) – “นักวิ่งคนสุดท้าย”

ในงานแข่งวิ่งท้องถิ่น มีนักวิ่ง N คน แต่ละคนใช้เวลาวิ่งจบสนามไม่เท่ากัน (เป็นจำนวนนาที)
คุณได้รับเวลาวิ่งของนักวิ่งทุกคนเรียงตามลำดับที่ลงทะเบียน
งานของคุณคือ หาว่าใครคือคนที่วิ่งช้าที่สุด (หมายเลขนักวิ่งเริ่มจาก 1)

Input Example

```
5
12 9 15 11 14
```

 Copy  Edit

Output Example

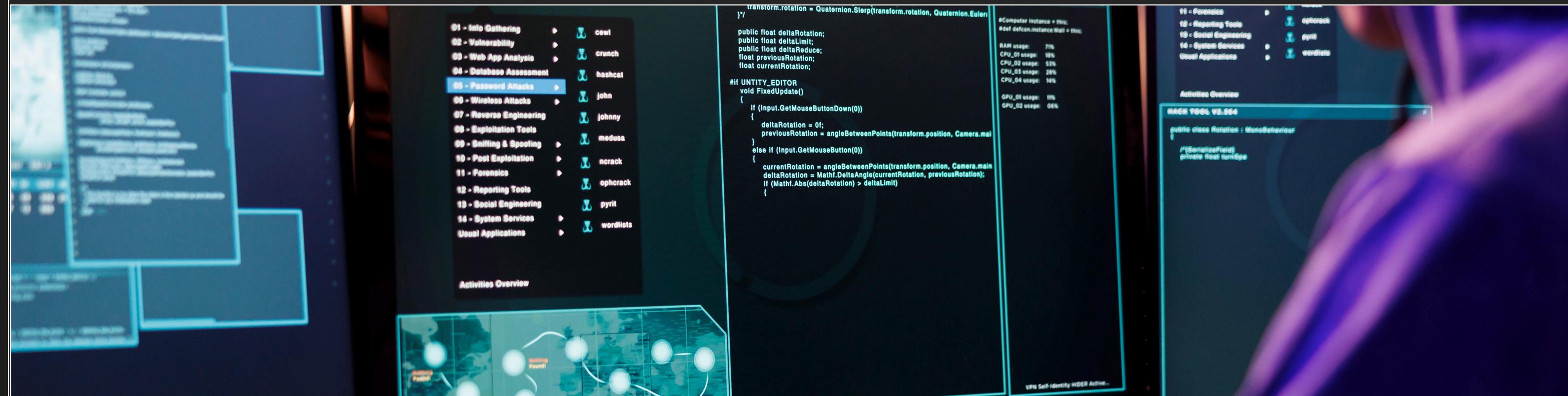
```
3
```

 Copy  Edit

(หมายเลข 3 คือคนที่วิ่งช้าที่สุด เพราะใช้เวลา 15 นาที)



Thankyou



WWW.REALLYGREATSITE.COM