



NHẬP MÔN LẬP TRÌNH

Hướng dẫn cài đặt và sử dụng CodeBlocks

Mục lục

1	Cài đặt	3
1.1	Download	3
1.2	Cài đặt	3
2	Viết chương trình C++ với CodeBlocks	10
2.1	Viết chương trình không cần tạo project	10
2.2	Viết chương trình trong một project	12
3	Debug chương trình C++ với CodeBlocks	20

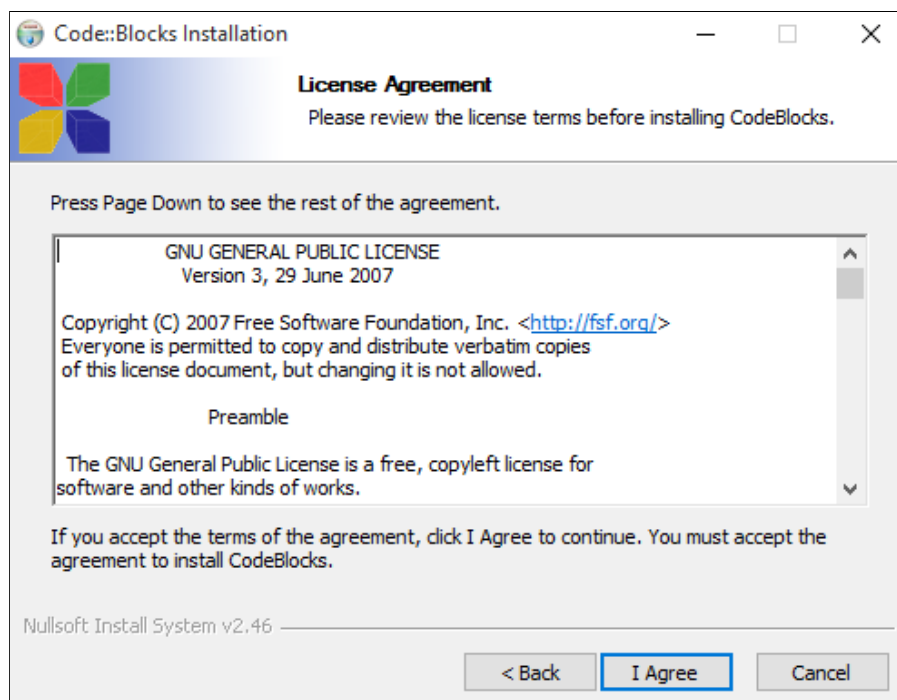
1 Cài đặt

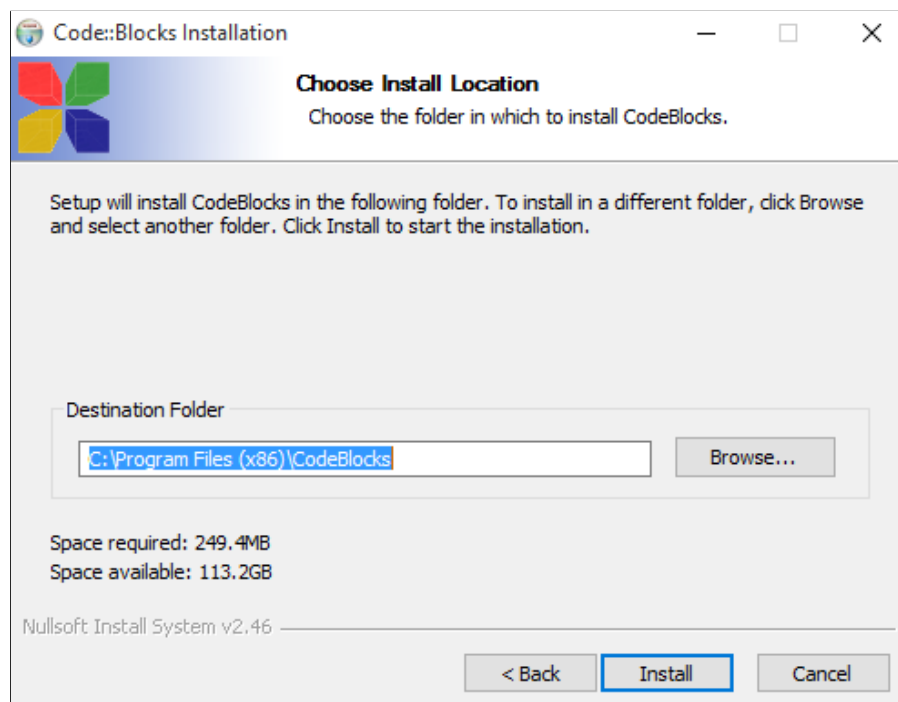
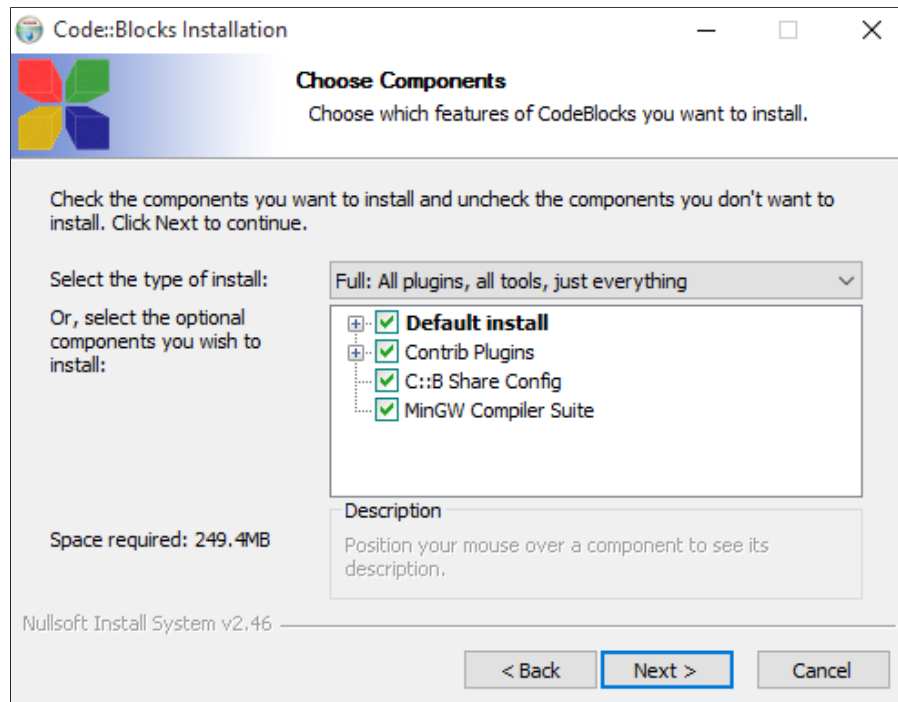
1.1 Download

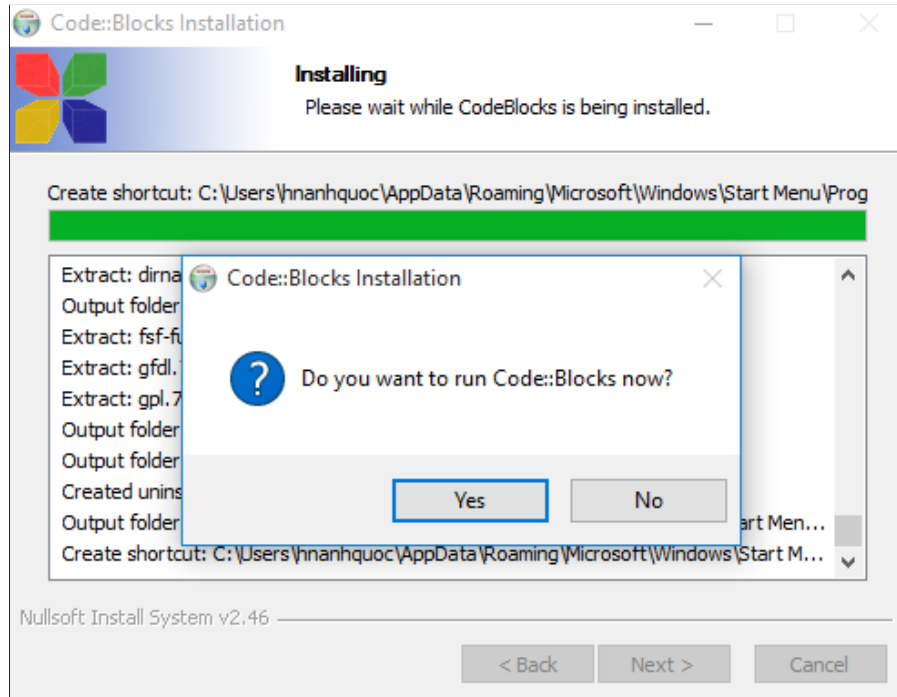
Download chương trình CodeBlocks 13.12 trên Windows tại [link](#).

1.2 Cài đặt

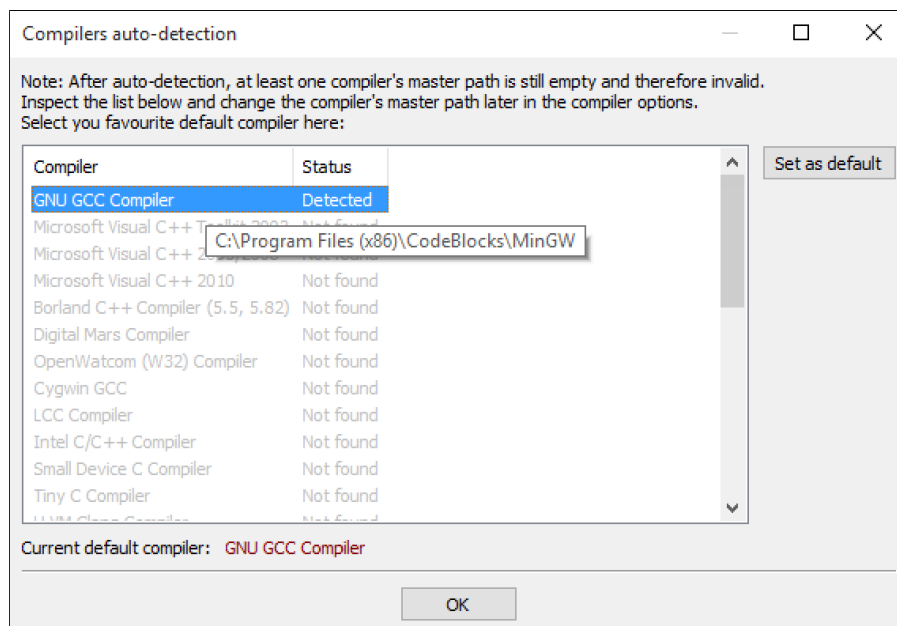
1. Chạy file vừa tải về (codeblocks-13.12mingw-setup.exe) với các cài đặt mặc định.



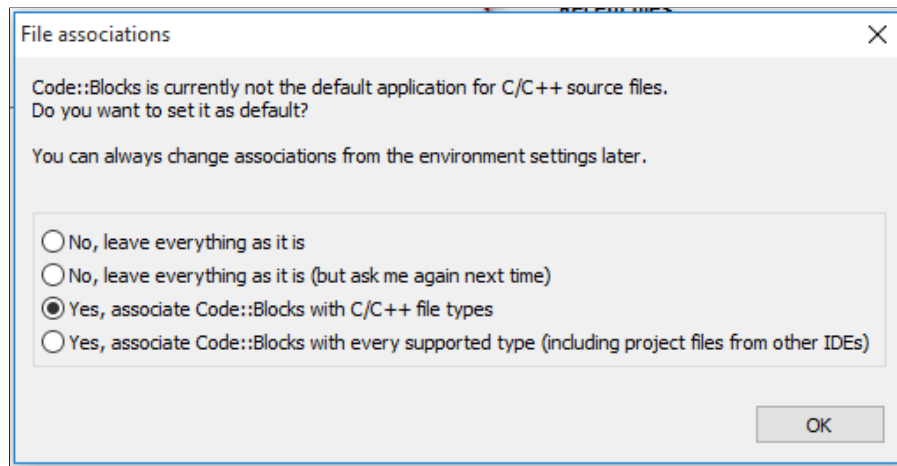




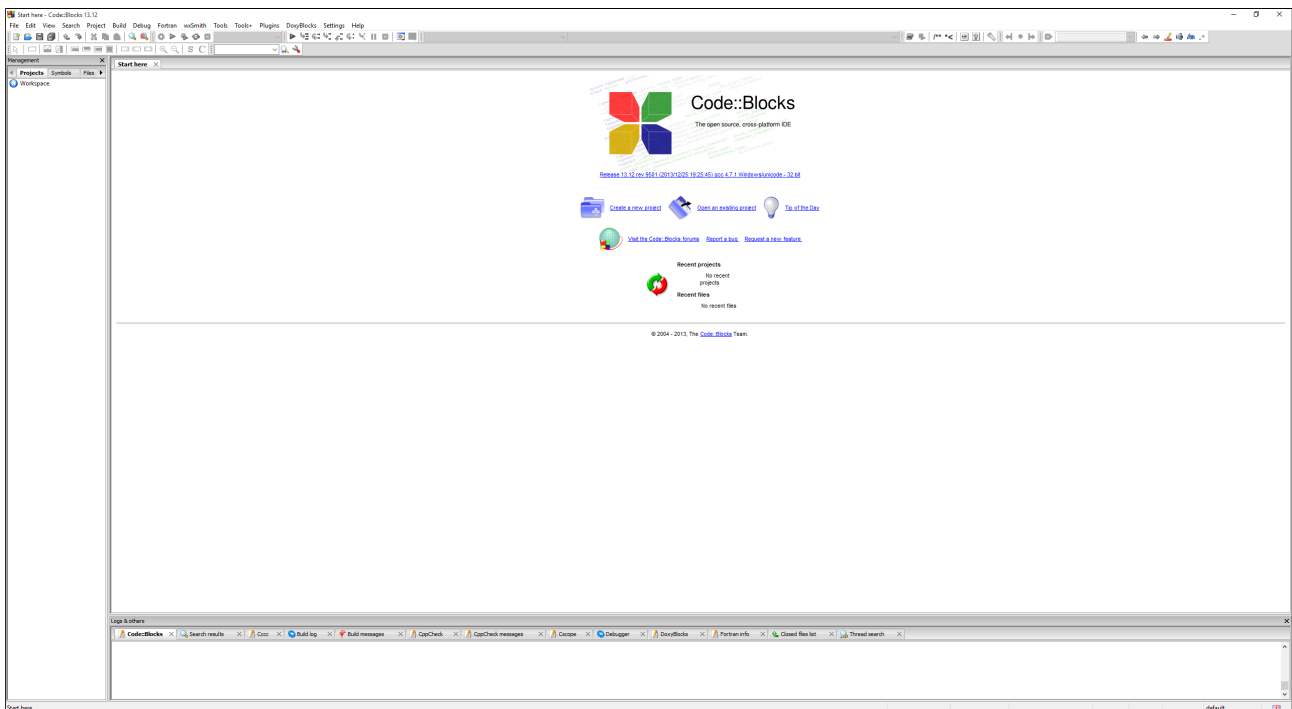
Chọn "Yes" để mở chương trình Code::Blocks. Sẽ xuất hiện thông báo để chọn trình biên dịch và chúng ta để mặc định (nếu như phía trên không có thay đổi đường dẫn).



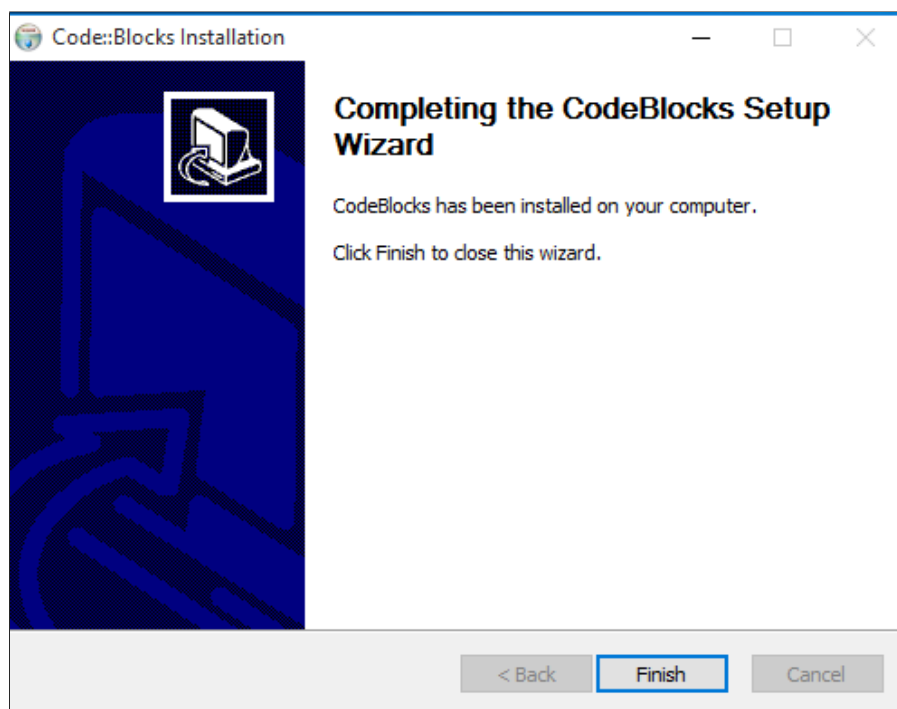
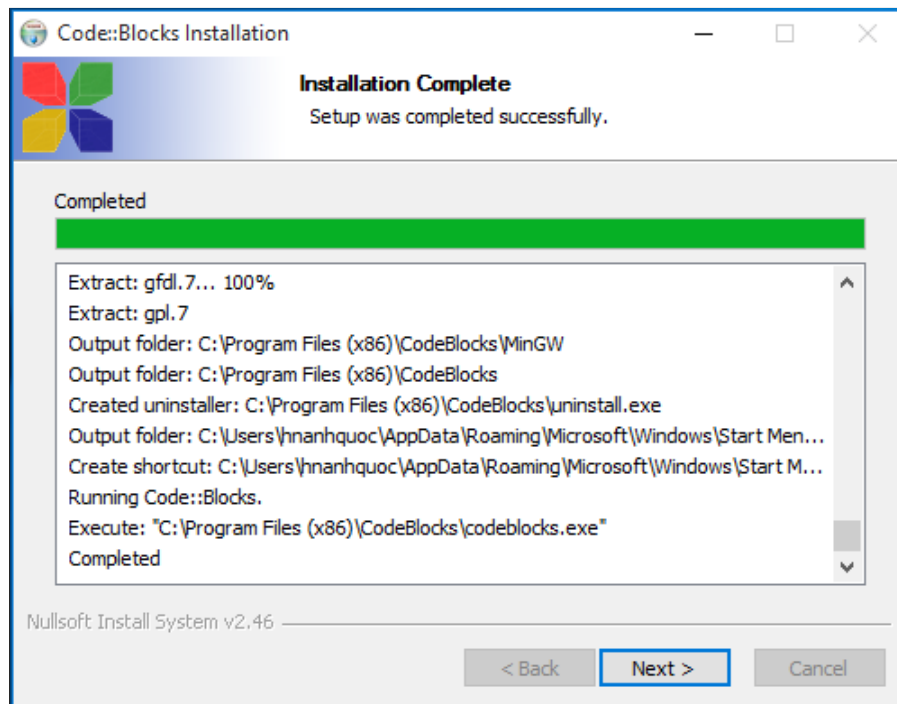
Xuất hiện 1 thông báo, hỏi chúng ta về việc sử dụng Code::Blocks cho các file định dạng .c và .cpp. Nên để mặc định và chọn "OK".



Chương trình sẽ có giao diện như sau:

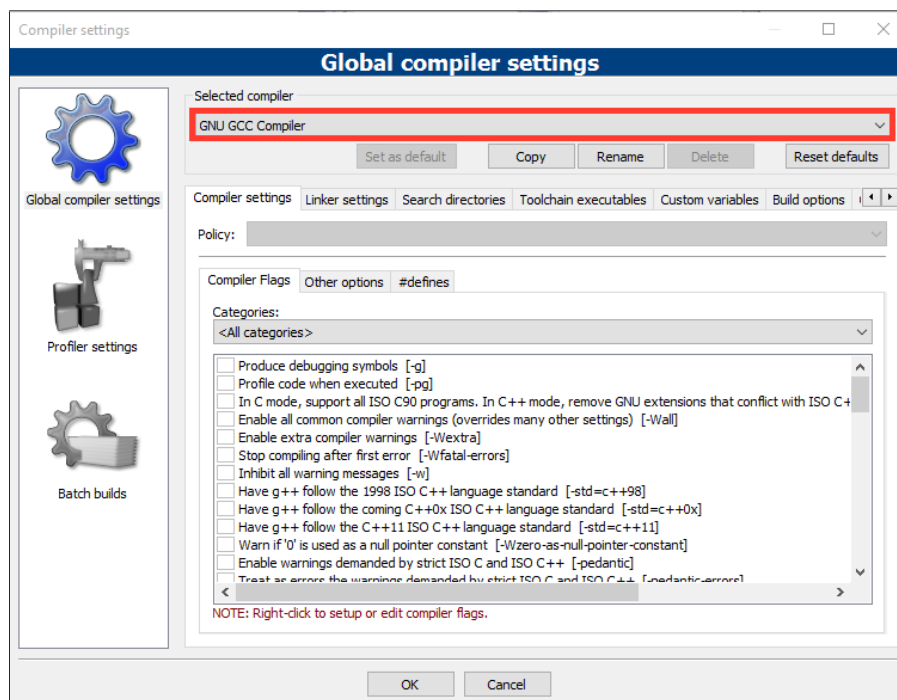
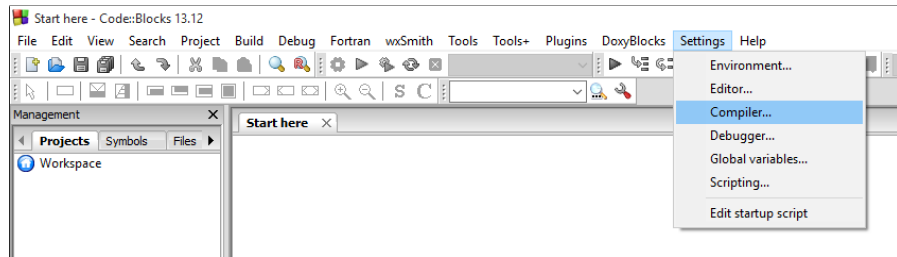


Sau khi chương trình đã mở, chúng ta quay lại trình cài đặt để hoàn tất quá trình cài đặt.

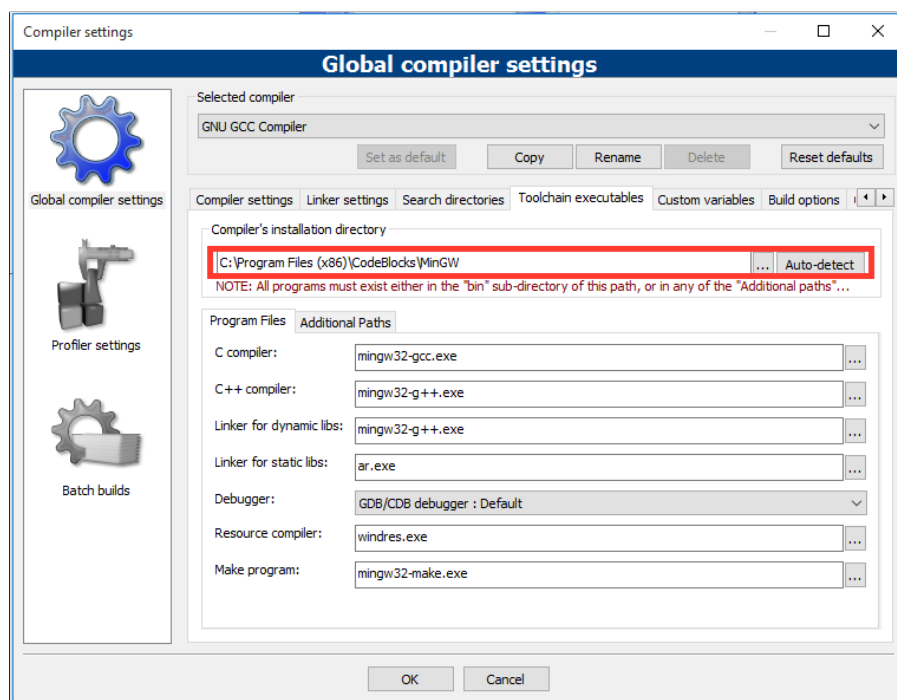


2. Kiểm tra các đường dẫn

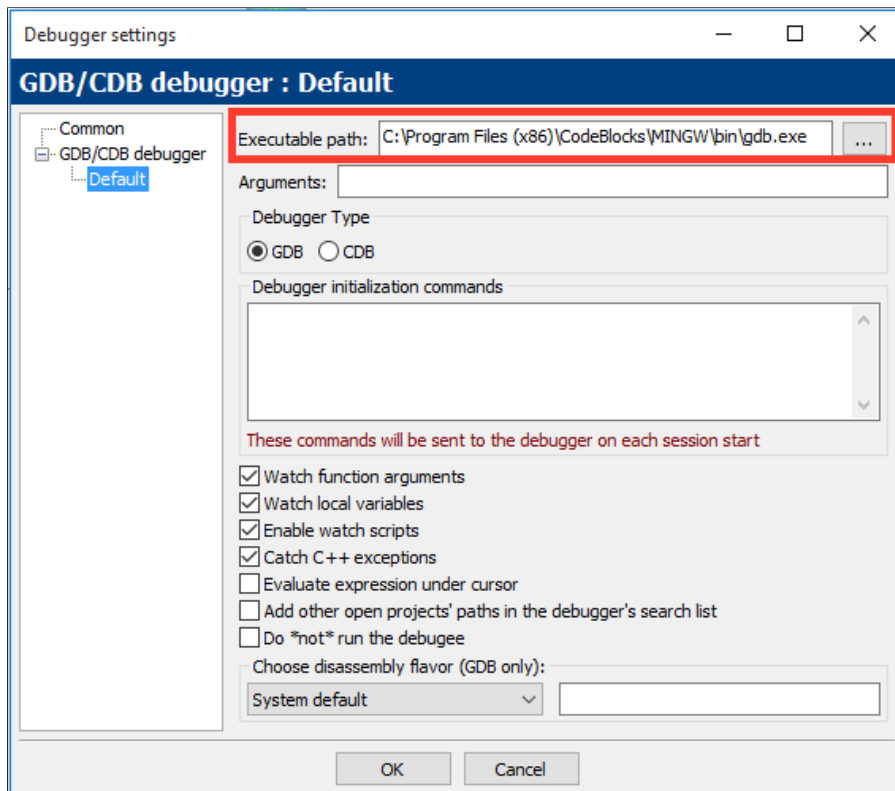
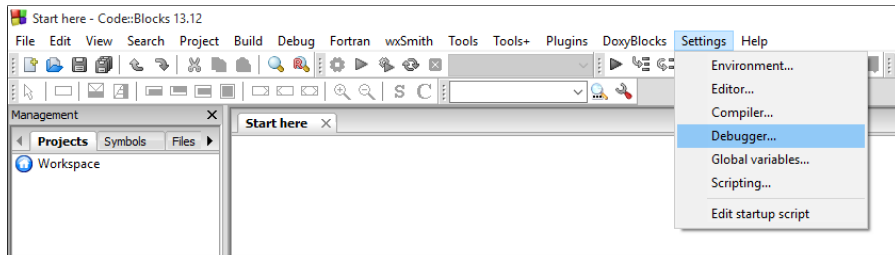
(a) Kiểm tra đường dẫn của Compiler



Chọn thẻ "Toolchain Executables".



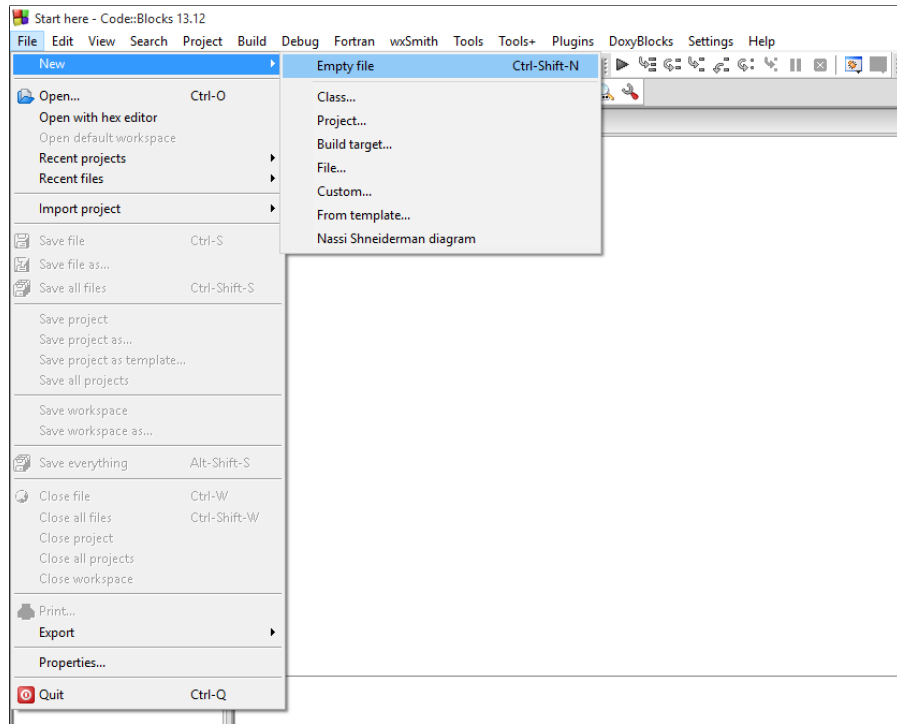
(b) Kiểm tra đường dẫn của Debugger



2 Viết chương trình C++ với CodeBlocks

2.1 Viết chương trình không cần tạo project

1. Tạo file

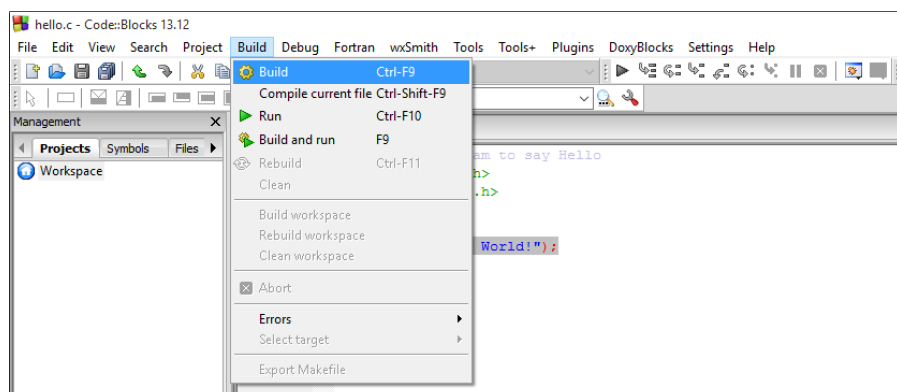


2. Nhập đoạn code sau:

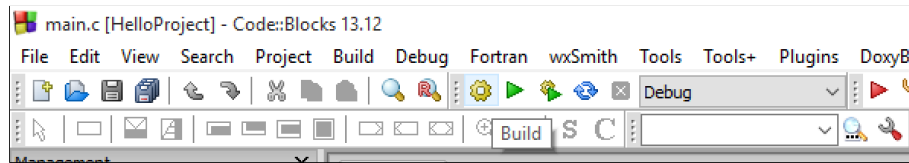
```
1 // First C program to say Hello
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 void main() {
6     printf("Hello World!");
7 }
8
```

3. Lưu file lại với tên "hello.c" ở một thư mục bất kì (có thể là "C:\Projects").

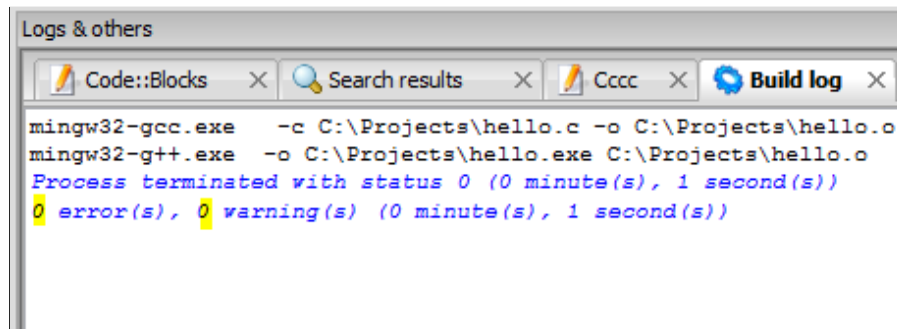
4. Build chương trình



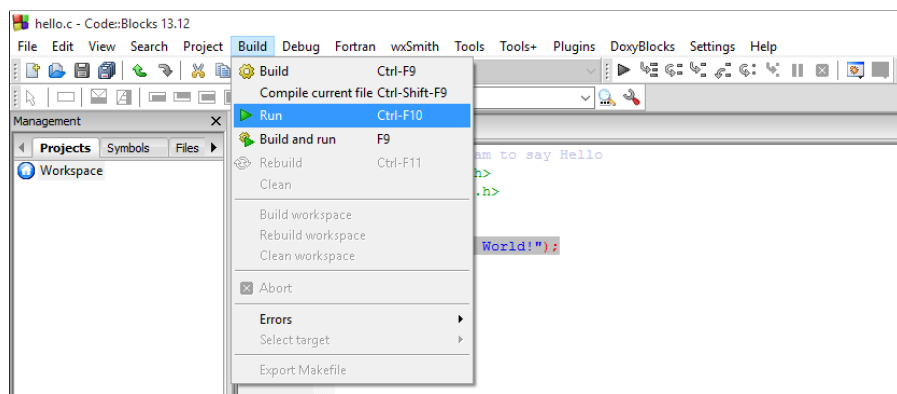
Hoặc sử dụng phím tắt trên thanh công cụ:



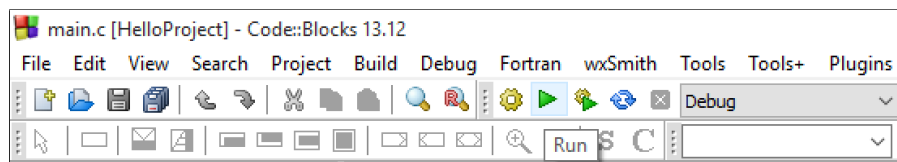
Chương trình Build thành công sẽ có thông báo như sau (bên dưới màn hình):



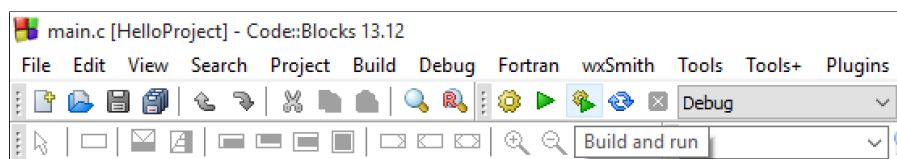
5. Chạy chương trình



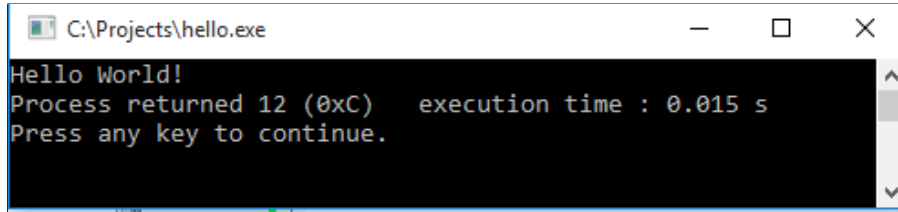
Tương tự, ta cũng có thể sử dụng phím tắt:



Hoặc



6. Kết quả

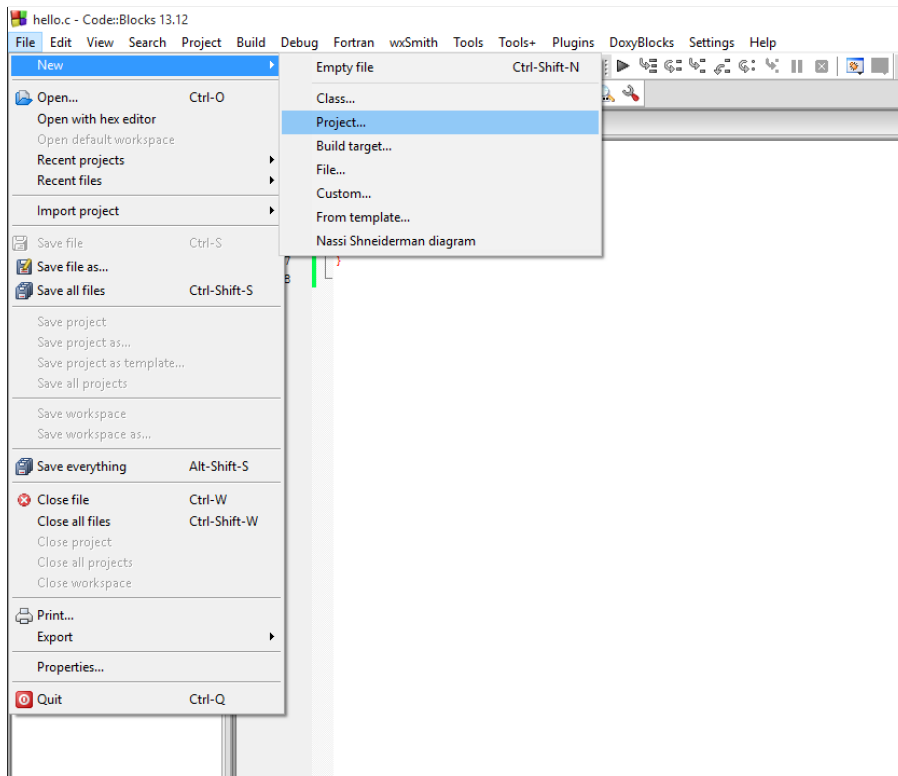


```
C:\Projects\hello.exe
Hello World!
Process returned 12 (0xC)  execution time : 0.015 s
Press any key to continue.
```

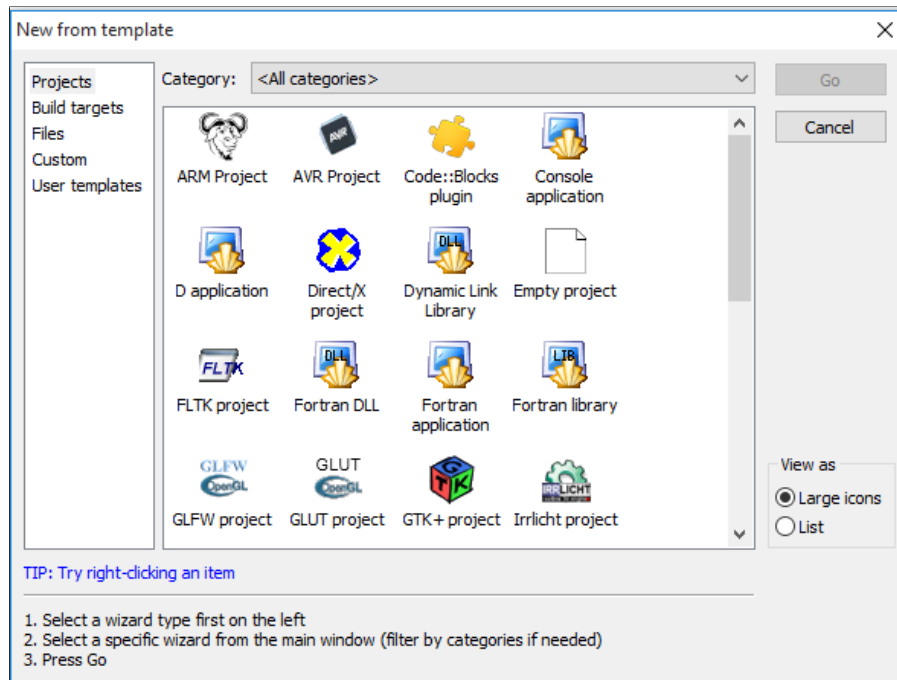
- **Ưu điểm:** Nhanh chóng và tiện lợi.
- **Nhược điểm:** Không thể debug khi có lỗi xảy ra. Để có thể debug, chúng ta sử dụng cách thứ 2: viết chương trình trong một project.

2.2 Viết chương trình trong một project

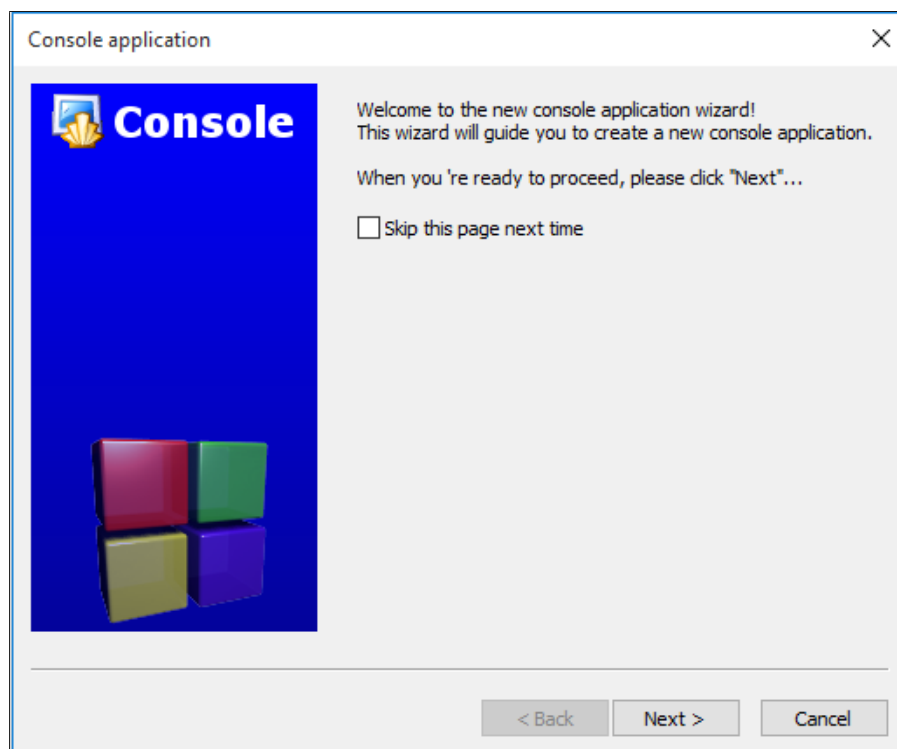
1. Tạo project mới



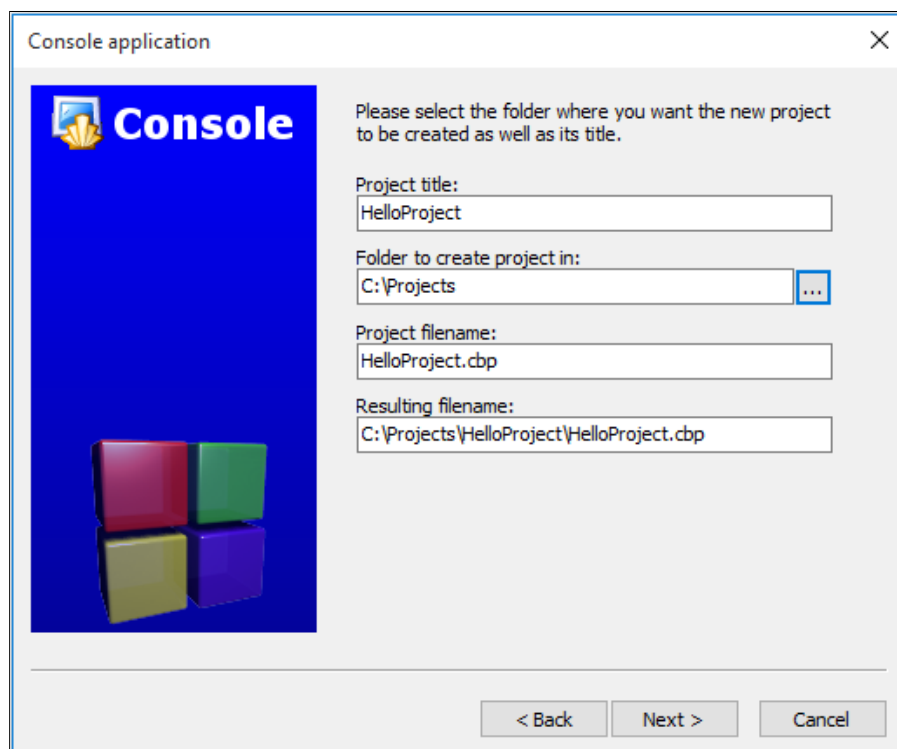
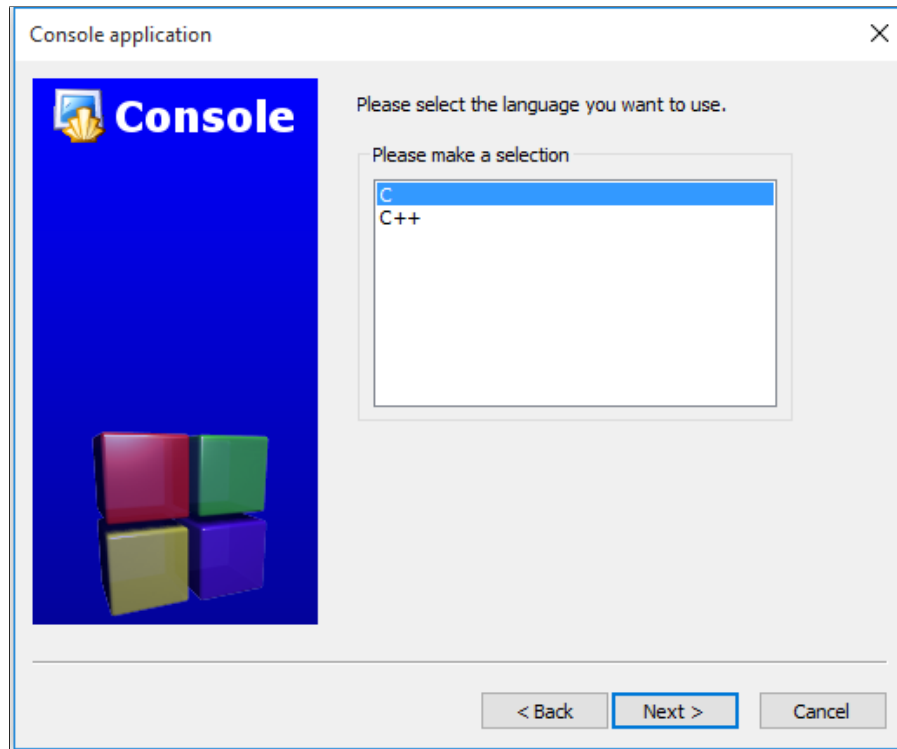
Chọn "Console Application", "Go"

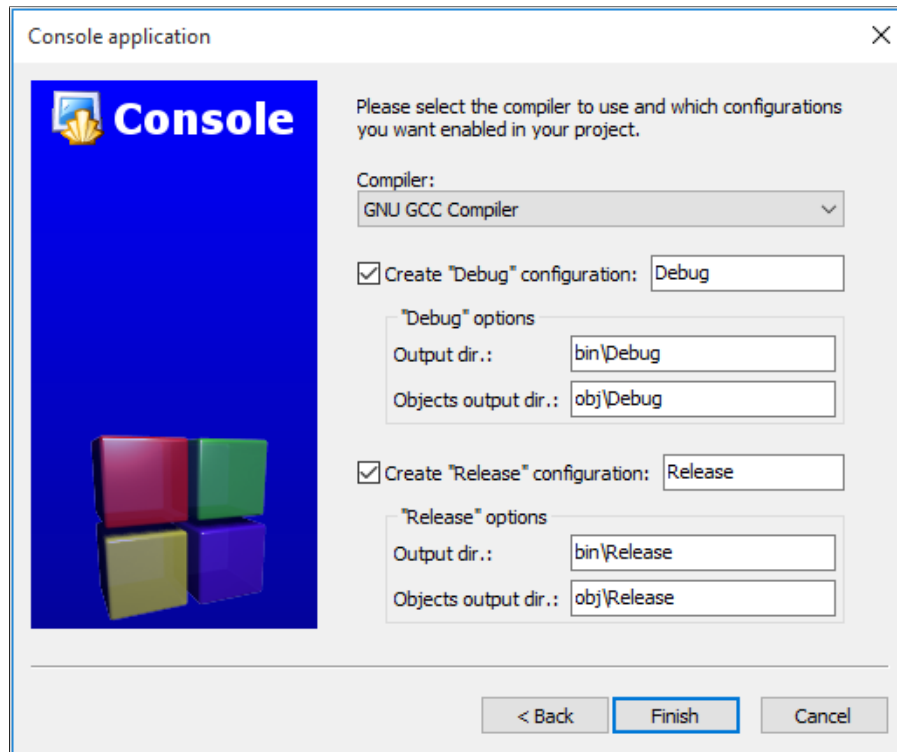


Nhấn "Next"

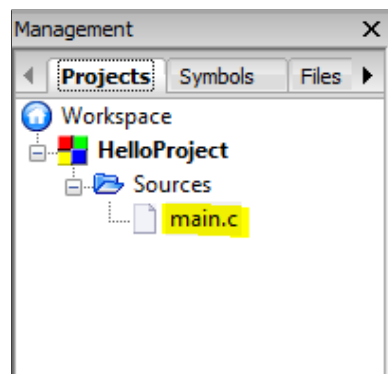


Chọn "C" và nhấn "Next"



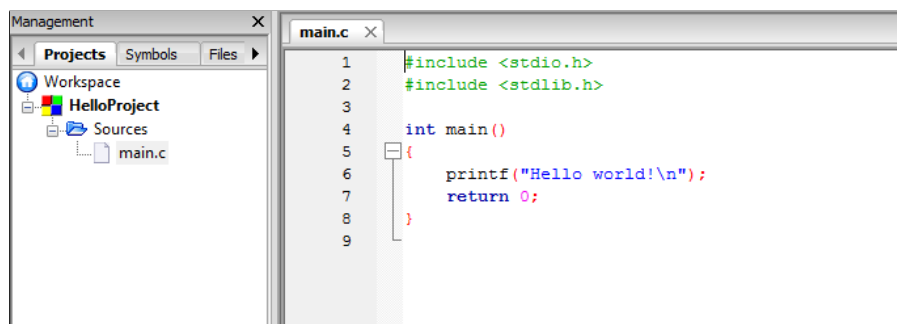


Sau khi hoàn tất, trên thanh "Management" nằm dọc bên trái sẽ xuất hiện project ta vừa tạo:



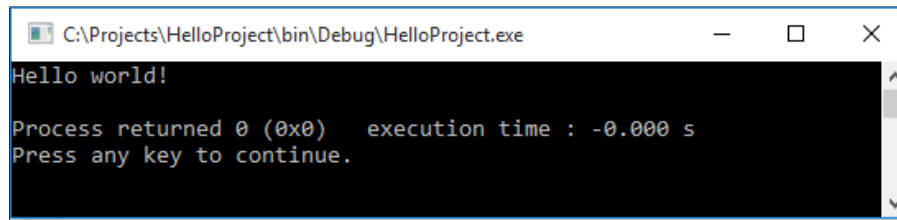
2. Source files

Nhấn đúp vào "main.c". Đây là code do chương trình tự sinh ra:



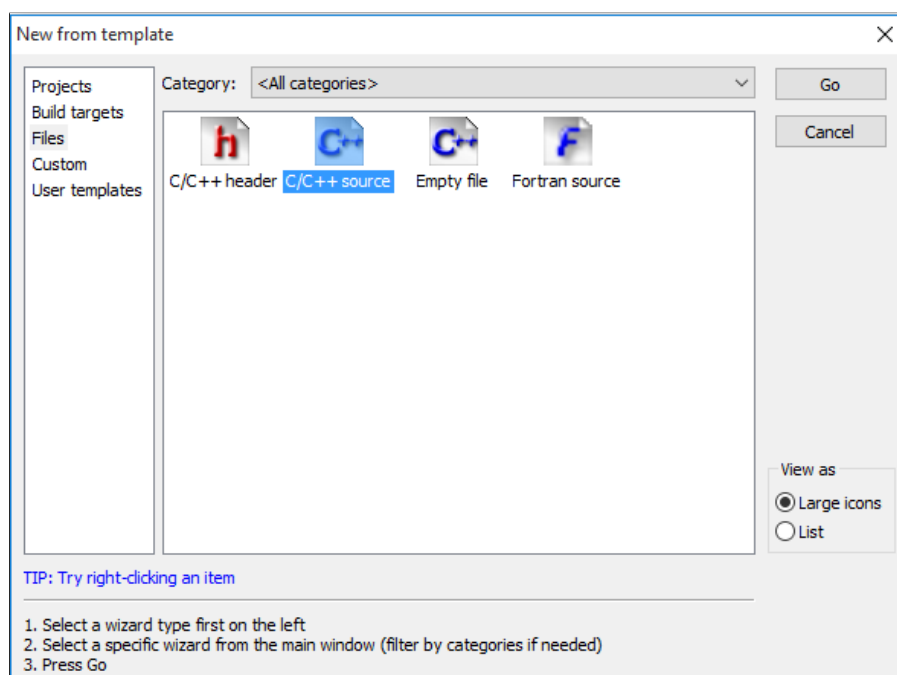
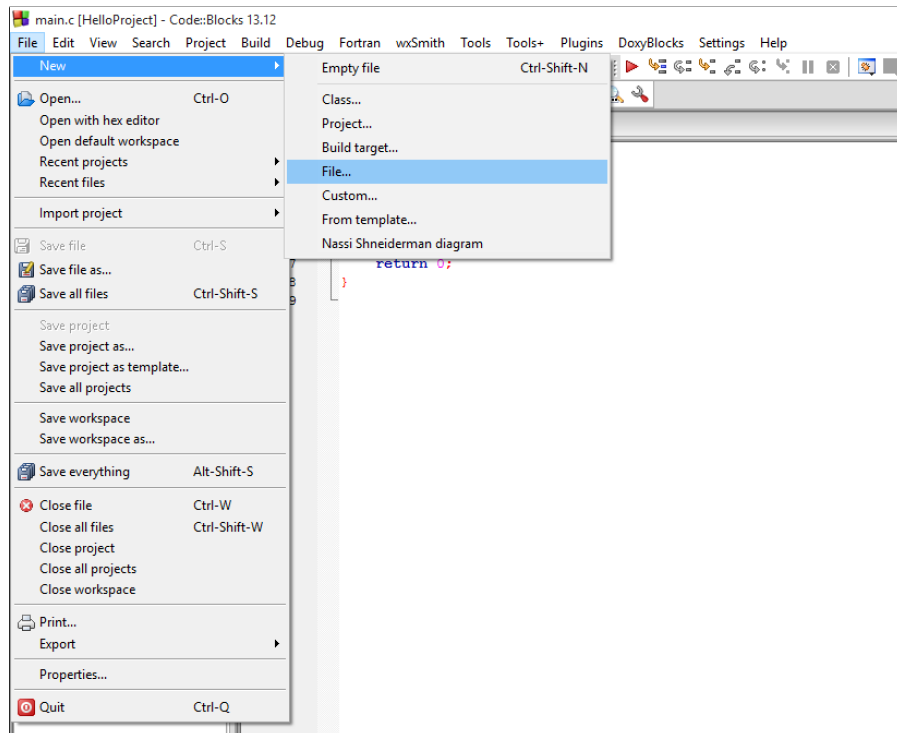
3. Build và chạy chương trình

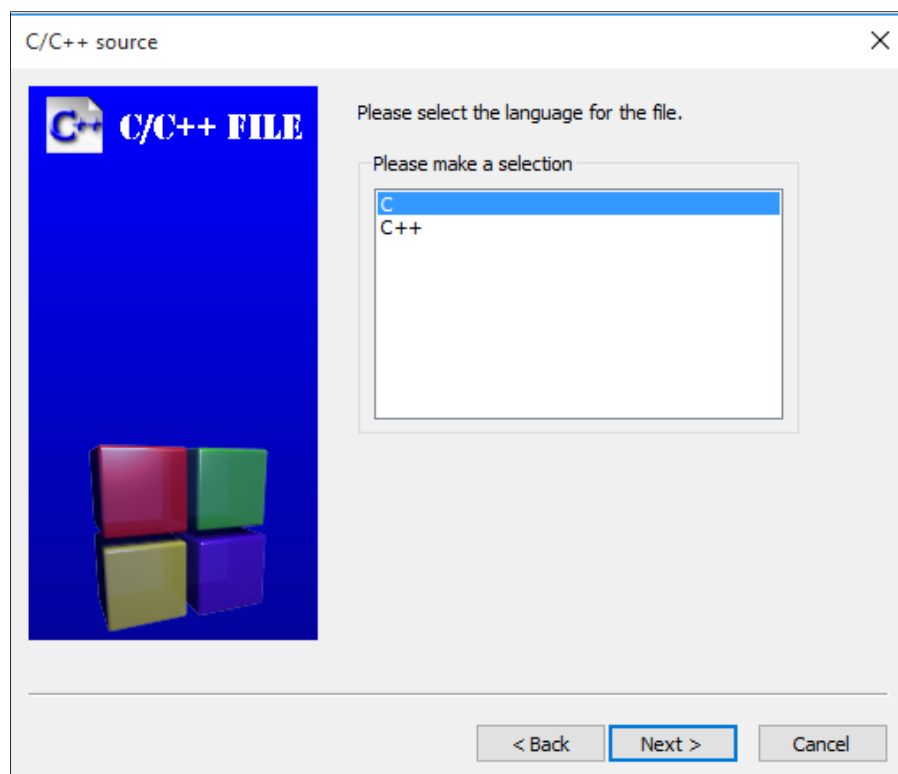
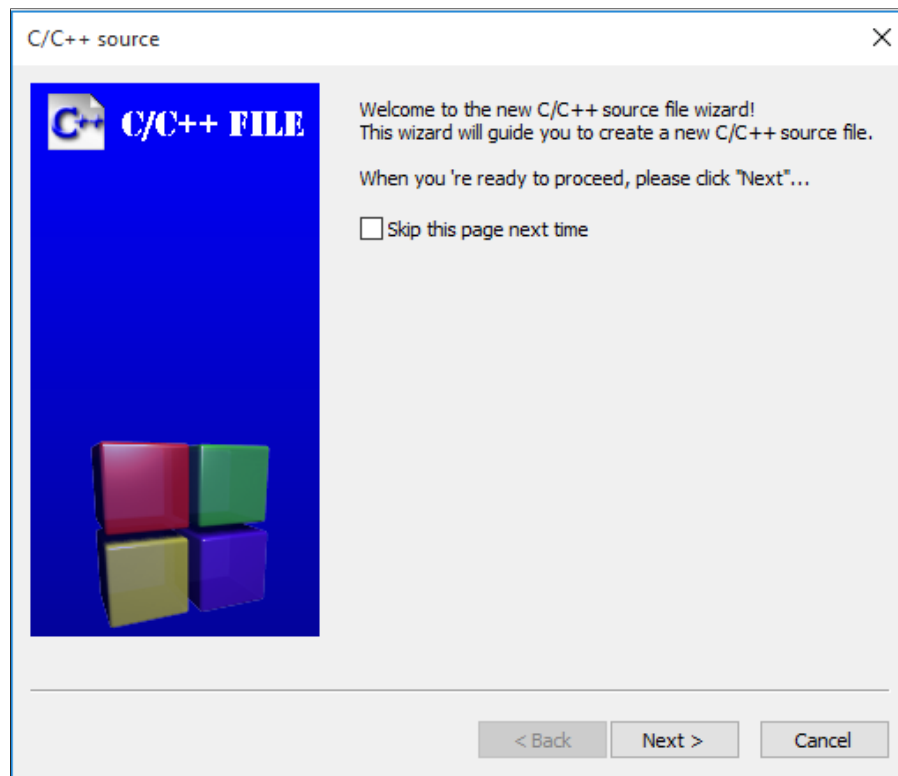
Tương tự như thao tác đối với viết chương trình không cần tạo project.



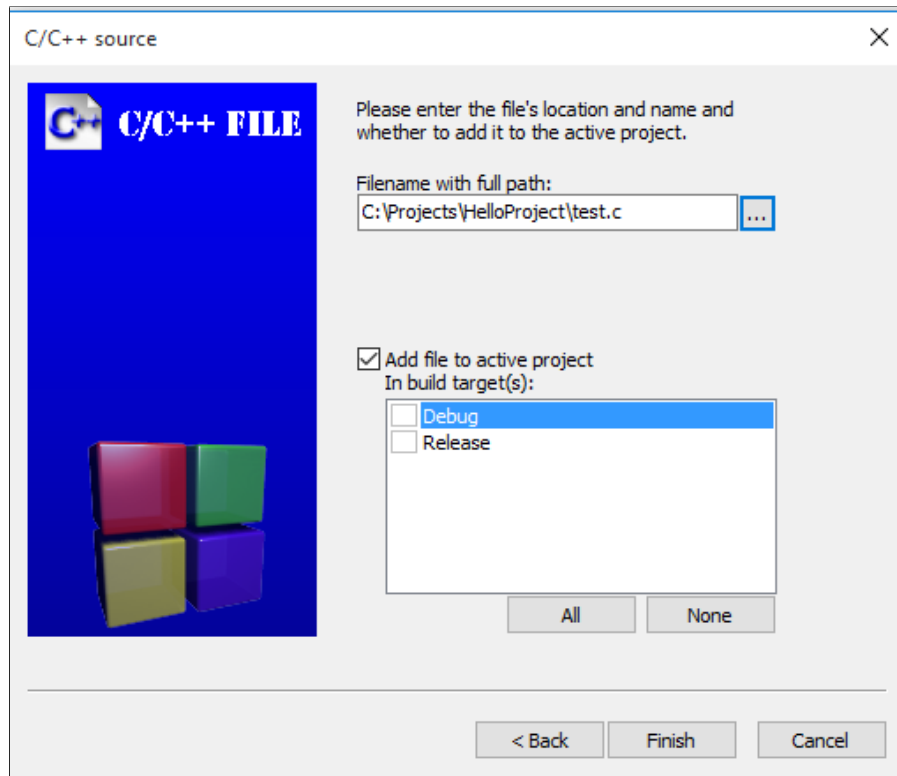
```
C:\Projects\HelloProject\bin\Debug\HelloProject.exe
Hello world!
Process returned 0 (0x0)   execution time : -0.000 s
Press any key to continue.
```

4. Thêm các source/header files

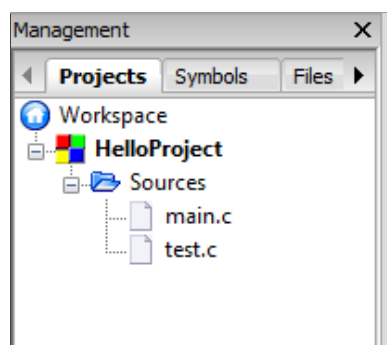




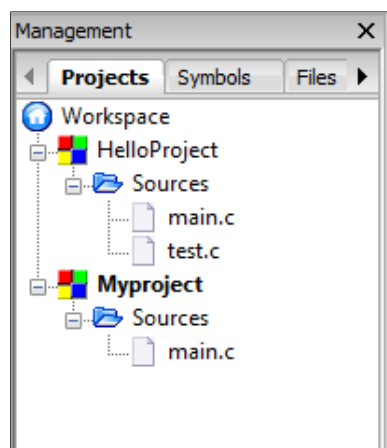
Chọn đường dẫn dẫn tới thư mục của project hiện tại và đặt tên:

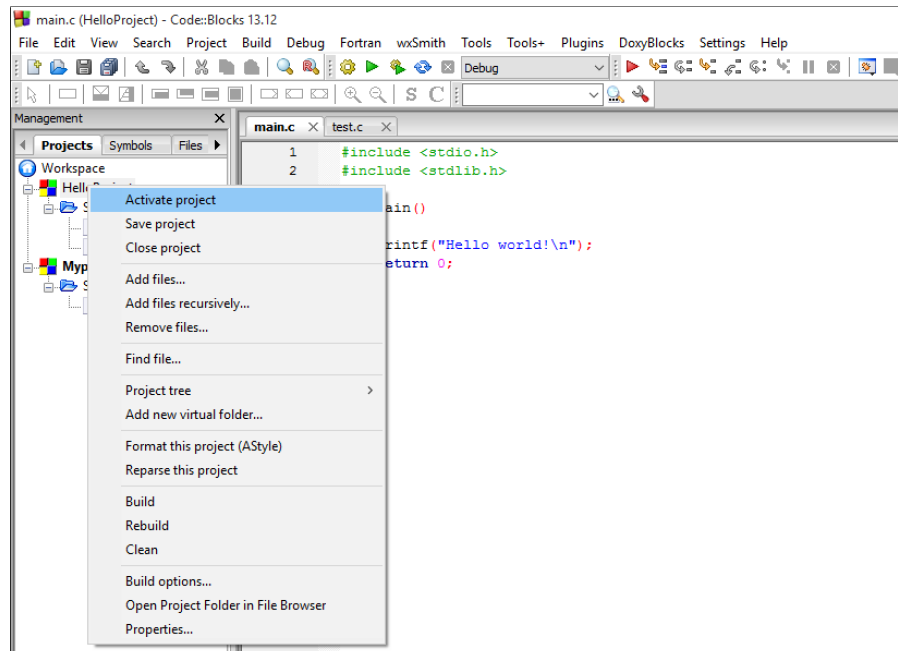


Kết quả:



Đối với trường hợp, chúng ta có nhiều projects, thì phải thực hiện **"Set Active"** cho project để Code::Blocks biết thao tác Build và Run là thực hiện cho project nào.





3 Debug chương trình C++ với CodeBlocks

1. Viết chương trình C

Nhập đoạn code sau vào file main.c:

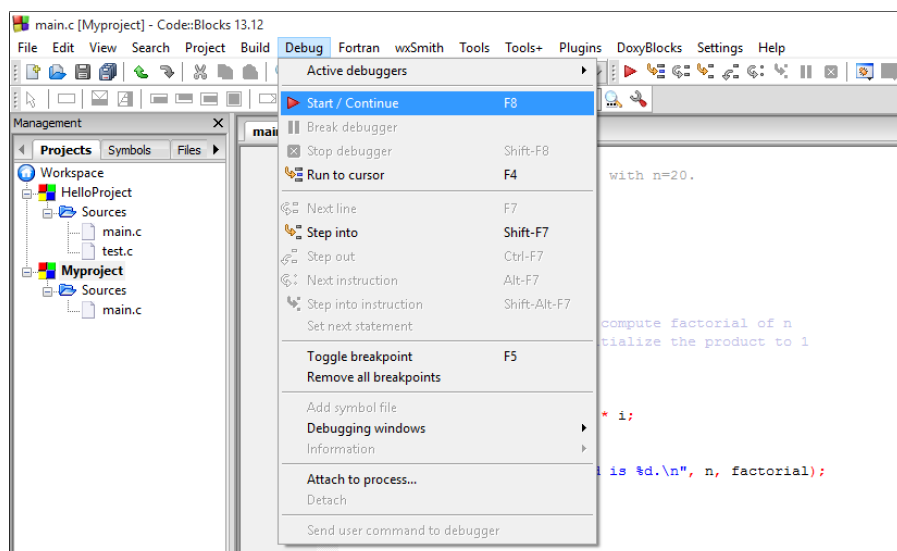
```
1  /*
2   * Compute the factorial of n, with n=20.
3   *   n! = 1*2*3*...*n
4   */
5  #include <stdio.h>
6  #include <stdlib.h>
7
8  int main()
9  {
10     int n = 20;        // To compute factorial of n
11     int factorial = 1; // Initialize the product to 1
12
13     int i = 1;
14     while (i <= n){
15         factorial = factorial * i;
16         i++;
17     }
18     printf("The Factorial of %d is %d.\n", n, factorial);
19     return 0;
20 }
21
```

2. Đặt các breakpoints

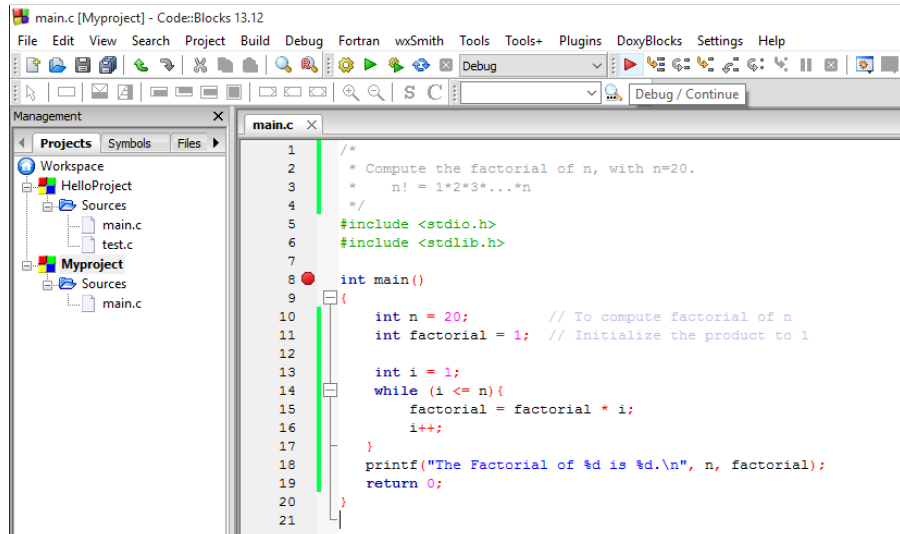
Breakpoint là các điểm mà chúng ta muốn chương trình dừng lại tại đó. Để đặt breakpoint, ta nhấn vào khoảng trống sau số dòng. Breakpoint được hiển thị là chấm màu đỏ, một chương trình có thể có 1 hoặc nhiều breakpoints.

```
7
8  ● int main()
9  {
10     int n = 20;        // To compute factorial of n
11     int factorial = 1; // Initialize the product to 1
12
```

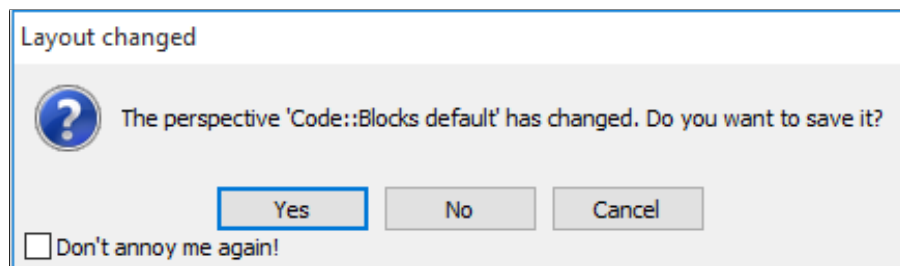
3. Chạy debug



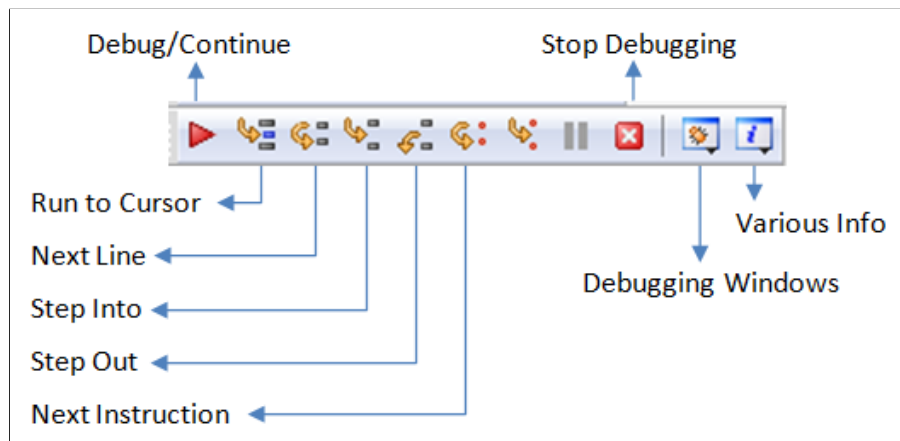
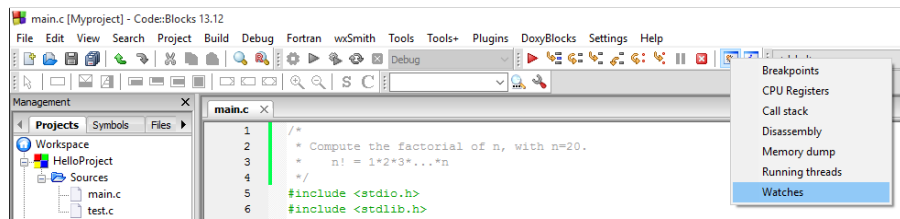
Hoặc



Một hộp thoại xuất hiện, hỏi rằng có muốn chuyển sang giao diện debug hay không? Chọn "Yes".



4. Bật các cửa sổ theo dõi



```
1  /*  
2  * Compute the factorial of n, with n=20.  
3  *   n! = 1*2*3*...*n  
4  */  
5  #include <stdio.h>  
6  #include <stdlib.h>  
7  
8  int main()  
9  {  
10     int n = 20;           // To compute factorial of n  
11     int factorial = 1;    // Initialize the product to 1  
12  
13     int i = 1;  
14     while (i <= n){  
15         factorial = factorial * i;  
16         i++;  
17     }  
18     printf("The Factorial of %d is %d.\n", n, factorial);  
19     return 0;  
20 }  
21
```

Function arguments		
Locals		
n	55	
factorial	47	
i	8	

```
1  /*  
2  * Compute the factorial of n, with n=20.  
3  *   n! = 1*2*3*...*n  
4  */  
5  #include <stdio.h>  
6  #include <stdlib.h>  
7  
8  int main()  
9  {  
10     int n = 20;           // To compute factorial of n  
11     int factorial = 1;    // Initialize the product to 1  
12  
13     int i = 1;  
14     while (i <= n){  
15         factorial = factorial * i;  
16         i++;  
17     }  
18     printf("The Factorial of %d is %d.\n", n, factorial);  
19     return 0;  
20 }  
21
```

Function arguments		
Locals		
n	20	
factorial	47	
i	8	

```
1  /*  
2  * Compute the factorial of n, with n=20.  
3  *   n! = 1*2*3*...*n  
4  */  
5  #include <stdio.h>  
6  #include <stdlib.h>  
7  
8  int main()  
9  {  
10     int n = 20;           // To compute factorial of n  
11     int factorial = 1;    // Initialize the product to 1  
12  
13     int i = 1;  
14     while (i <= n){  
15         factorial = factorial * i;  
16         i++;  
17     }  
18     printf("The Factorial of %d is %d.\n", n, factorial);  
19     return 0;  
20 }  
21
```

Function arguments		
Locals		
n	20	
factorial	-2102132736	
i	21	

Quan sát giá trị các biến có thể giúp ta phát hiện được lỗi của chương trình.