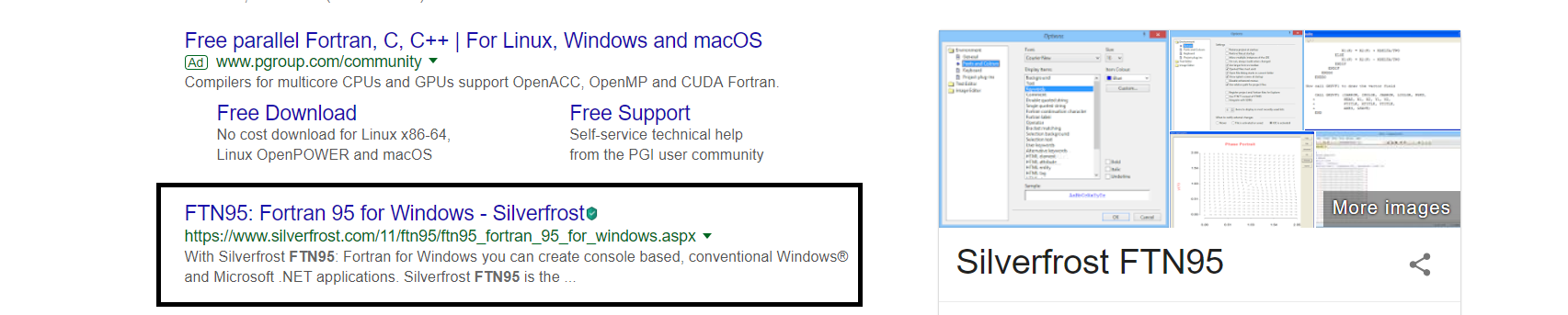
1. **Lịch sử hình thành FORTRAN 90**

Tập hợp các qui tắc đặc biệt để mã hóa những kiến thức cho máy tính hiểu được gọi là ngôn ngữ lập trình. Có rất nhiều ngôn ngữ như vậy, ví dụ: FORTRAN, BASIC, Pascal, C ,... FORTRAN là tên cấu tạo từ FORmula TRANslation (diễn dịch công thức, hay còn gọi là công thức dịch), là một trong những ngôn ngữ lập trình bậc cao đầu tiên. Nó có thể sử dụng những tên tượng trưng để biểu diễn định lượng toán học và viết các công thức toán học dưới dạng định thức hợp lý có thể hiểu được như: X = (-B+DELTA)/(2\*A).

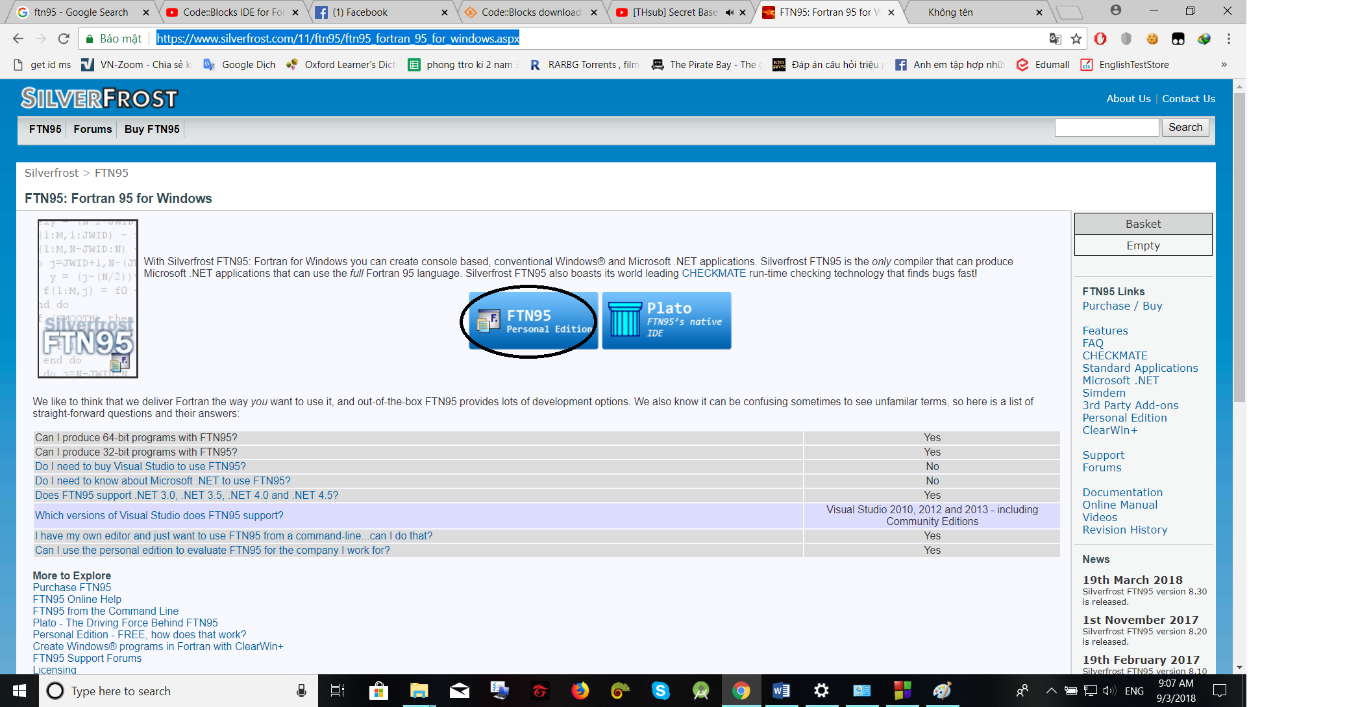
Ý tưởng của FORTRAN được John Backus đề xuất vào khoảng cuối năm 1953 ở New York, và chương trình FORTRAN đầu tiên đã được chạy vào tháng 4 năm 1957. Kể từ đó, việc sử dụng FORTRAN đã nhanh chóng được phổ biến rộng rãi. Điều đó đòi hỏi sự cần phải sớm tiêu chuẩn hóa nó sao cho chương trình viết ra phải đảm bảo chạy được ở mọi nơi. Vào năm 1966, lần đầu tiên phiên bản chuẩn của ngôn ngữ lập trình này được ấn hành. Phiên bản này, như đã biết, đó là FORTRAN 66. Phiên bản chuẩn mới sau đó, FORTRAN 77 được ấn hành vào năm 1978. Tuy nhiên các phiên bản Fortran này được phát triển ban đầu như là một ngôn ngữ thủ tục. Không bằng lòng với sự cạnh tranh của các ngôn ngữ mới khác, như Pascal và C, vì thế FORTRAN đã tiếp tục phát triển một cách mạnh mẽ. Và phiên bản chuẩn gần đây, FORTRAN 90, với nhiều đặc tính đột phá khi có thêm các tính năng hỗ trợ về lập trình hướng đối tượng như đệ qui, con trỏ, modul, local routines, và một số tính năng về ma trận đã ra đời vào tháng 8 năm 1991. Cho đến nay, FORTRAN đã phát triển đến những phiên bản mới hơn, như FORTRAN 95, FORTRAN 2003. Tuy nhiên chúng ta sẽ chỉ trình bày về phiên bản FORTRAN 90. Vì những phần bổ sung của các phiên bản sau là không nhiều so với FOTRAN 90 và chưa cần thiết để chúng ta xét đến.

1. **Hướng dẫn cài đặt và sử dụng**
2. **SỬ DỤNG COMPILER FTN95**
   1. Cài đặt

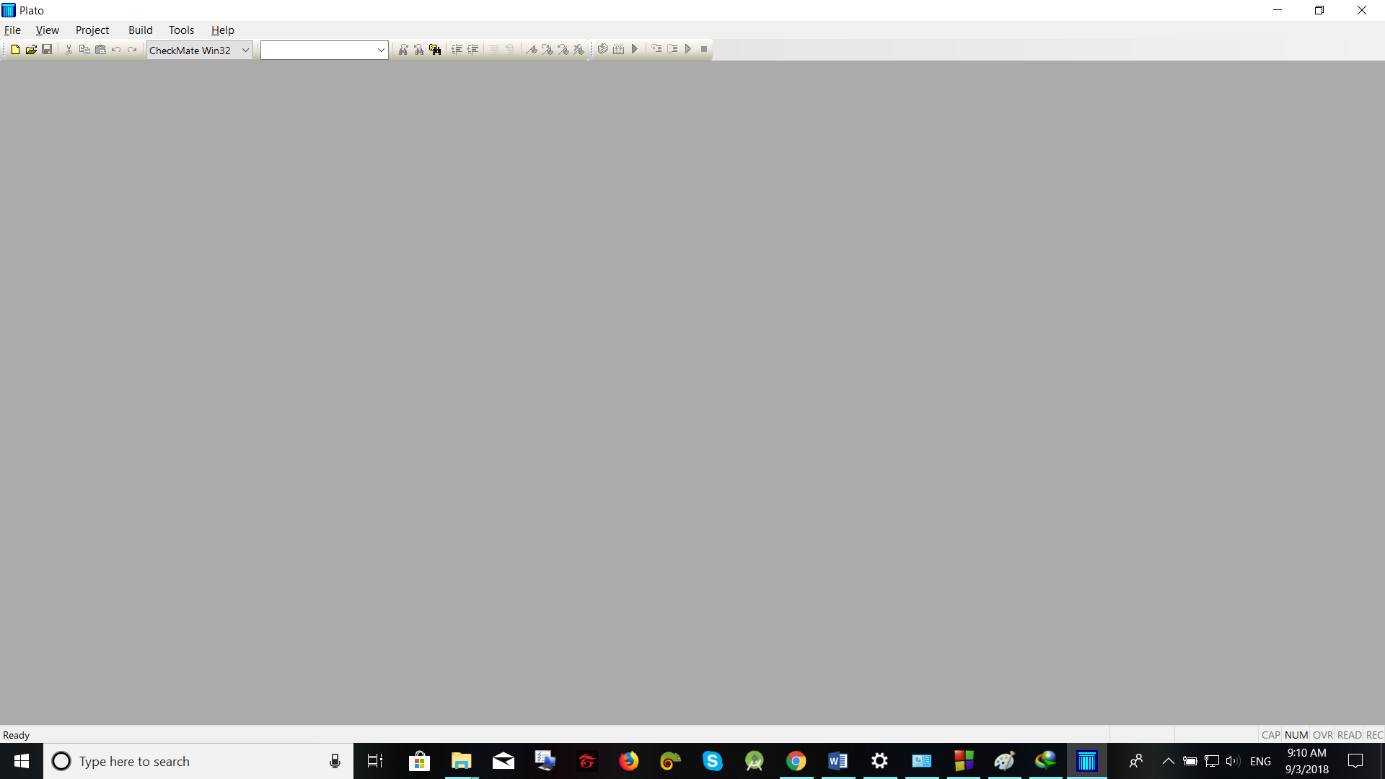
bước 1: truy cập google.com search: ftn95

bước 2:click vào link:

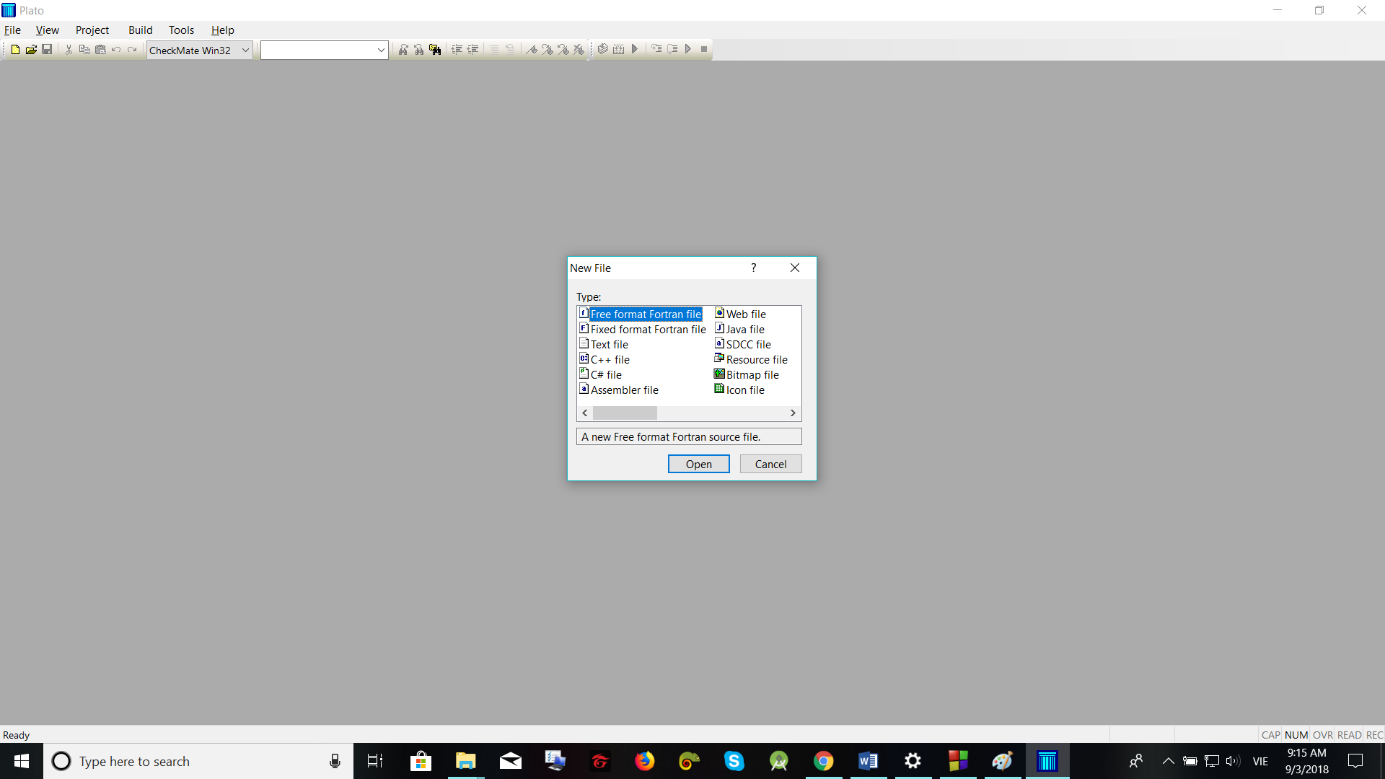
<https://www.silverfrost.com/11/ftn95/ftn95_fortran_95_for_windows.aspx>



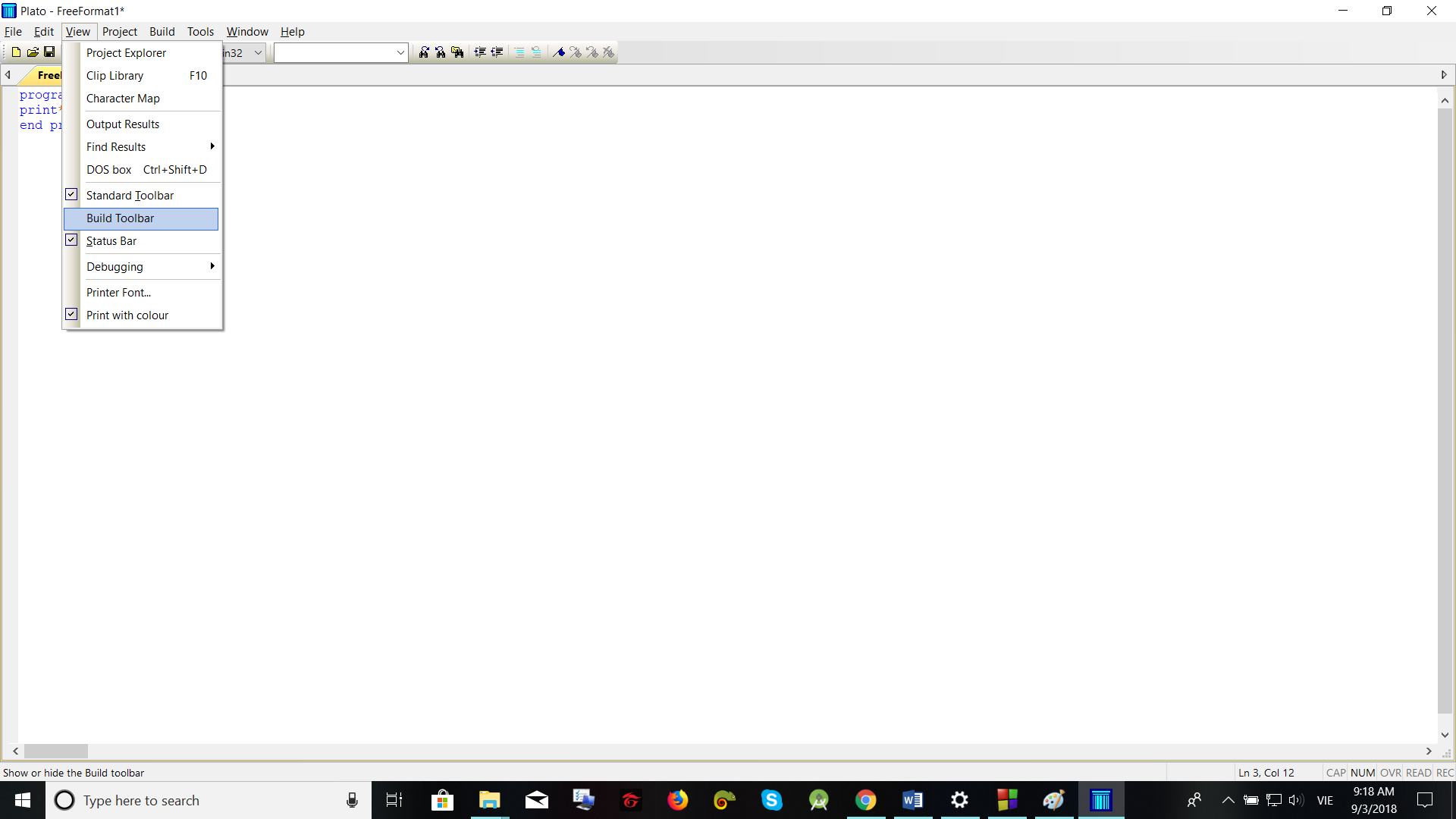
Bước 3: chọn ftn95 persional edittion và dowload về và cài đặt

**Giao diện chuuơng trình**

* 1. Sử dụng

Chọn file -> new or ctrl + n để tạo file mới -> free format fortran file -> open

Chọn view-> build toolbar để hiện thị tool hỗ trợ



****Sau khi chọn thanh công cụ sẽ hiện ra:

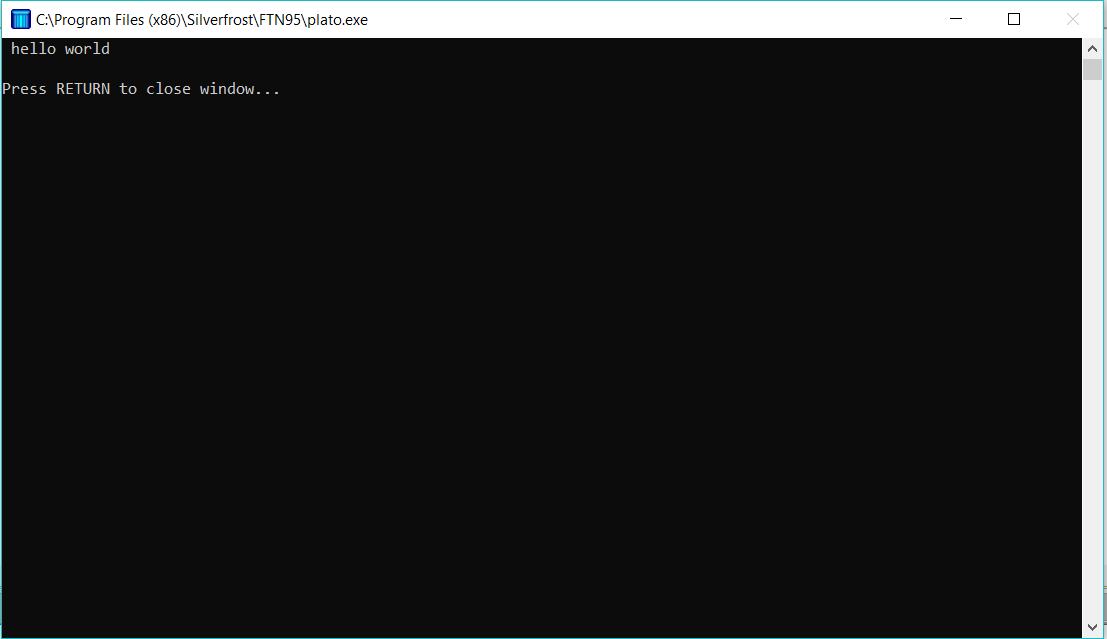
Ta tiến hành viết thử một đoạn chương trình helloworld

**program helloworld**

**print\*, 'hello world'**

**end program**

clicik nút start hoặc dùng ctrl + f5 để chạy chương trình

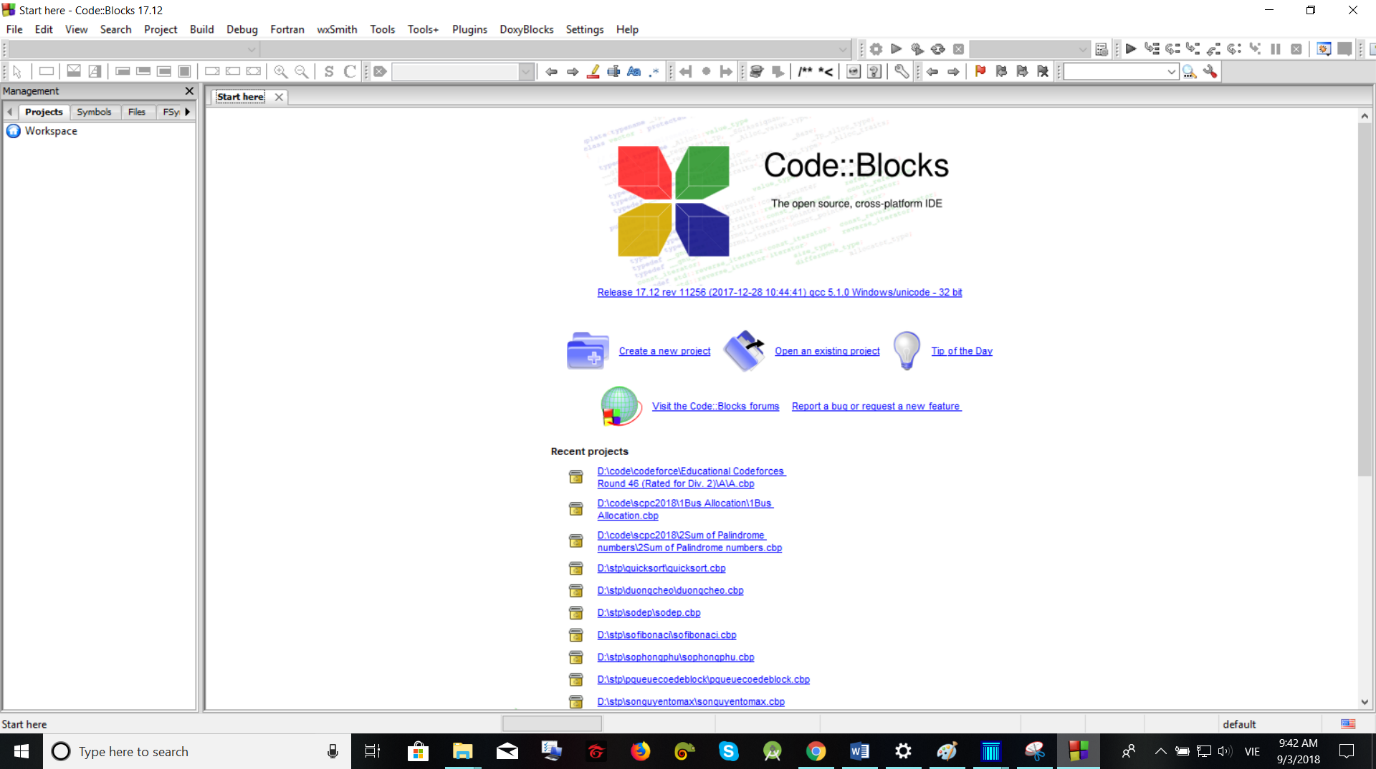
kết quả đạt được

1. **Sử dụng ide codeblocks**
   1. cài đặt

Bước 1: truy cập codebocks.org ->dowload -> binaries -> codeblocks-17.12mingw\_fortran-setup.exe

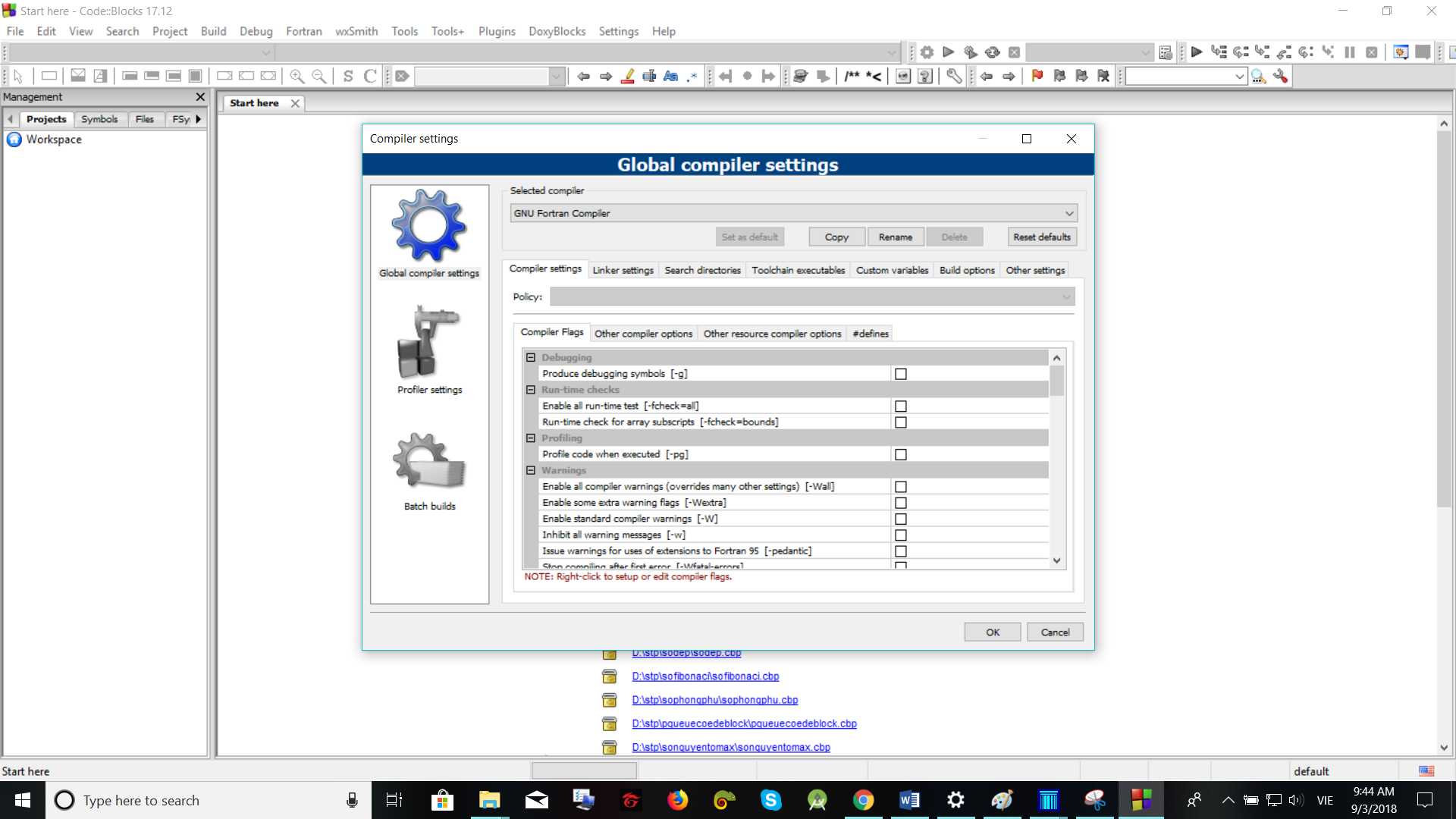
Bước 2: sau khi tải về máy ta tiến hành cài đặt



Chọn next -> i agree -> next ->install, sau khi cài đặt ta tiến hành mở chương trình

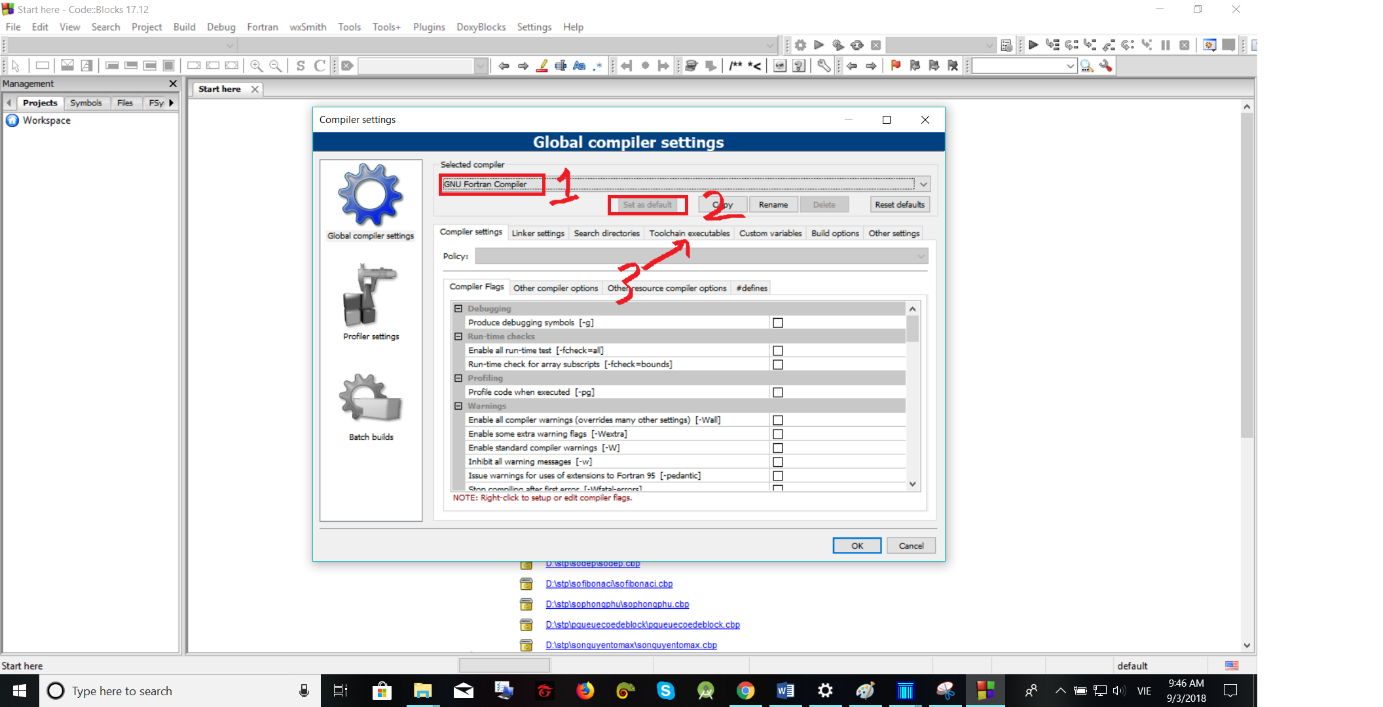
Chú ý: với những bạn đã cài codeblocks nếu đã tích hợp gnu fortran có thể bỏ qua bước trên và chỉ setup một vài thông số

Chọn setting -> compiler

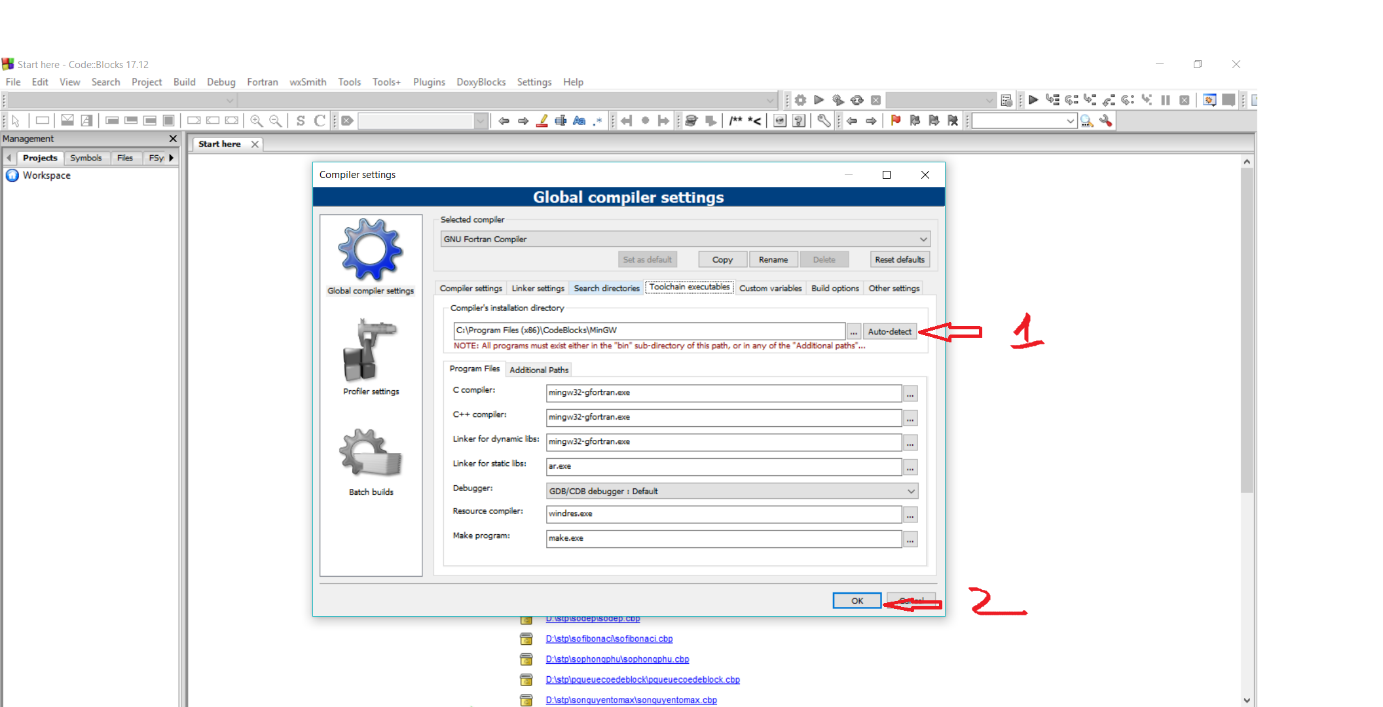


Tại select compiler chọn GNU Fortran compiler -> set a default.

Sau đó chọn Toolchain executables



Tại compiler installation directory chọn auto detect để code block tự động tìm gnu fortran -> ok



Vậy là đã setup xong.

* 1. Sử dụng

-tạo một chương trình helloworld

Chọn file-> new or ctrl + shift + n

* Tiến hành viết code

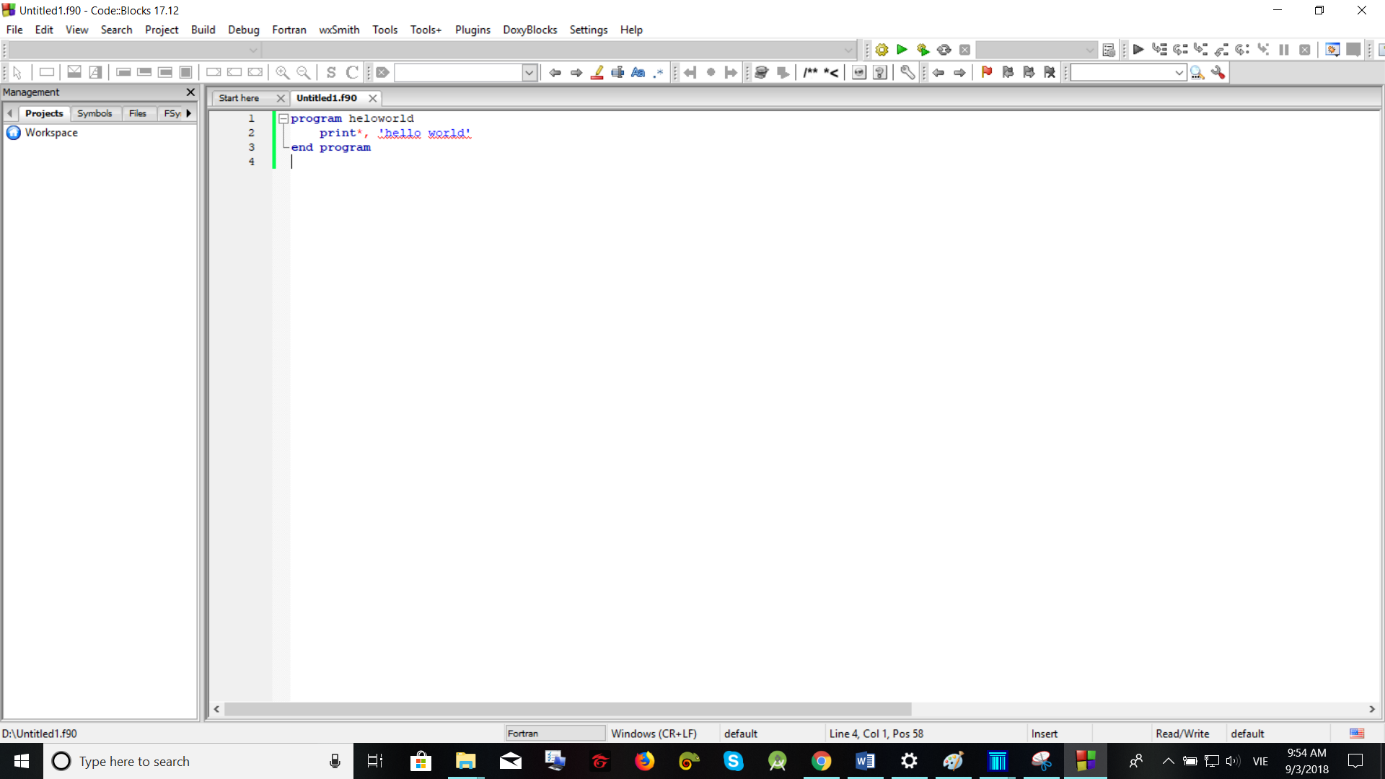
**program heloworld**

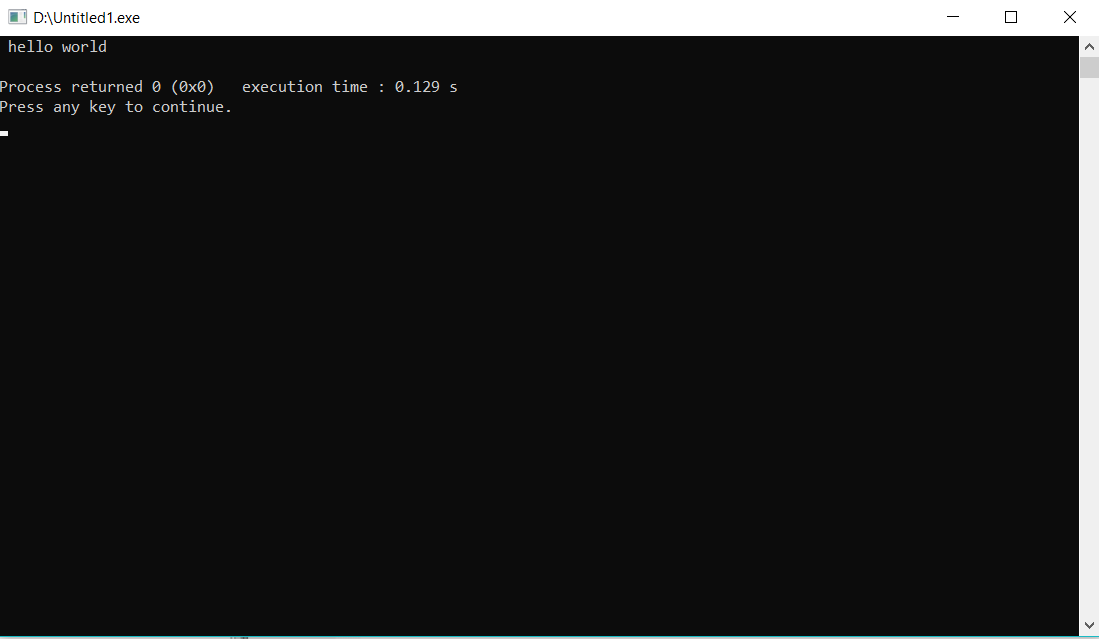
**print\*, 'hello world'**

**end program**

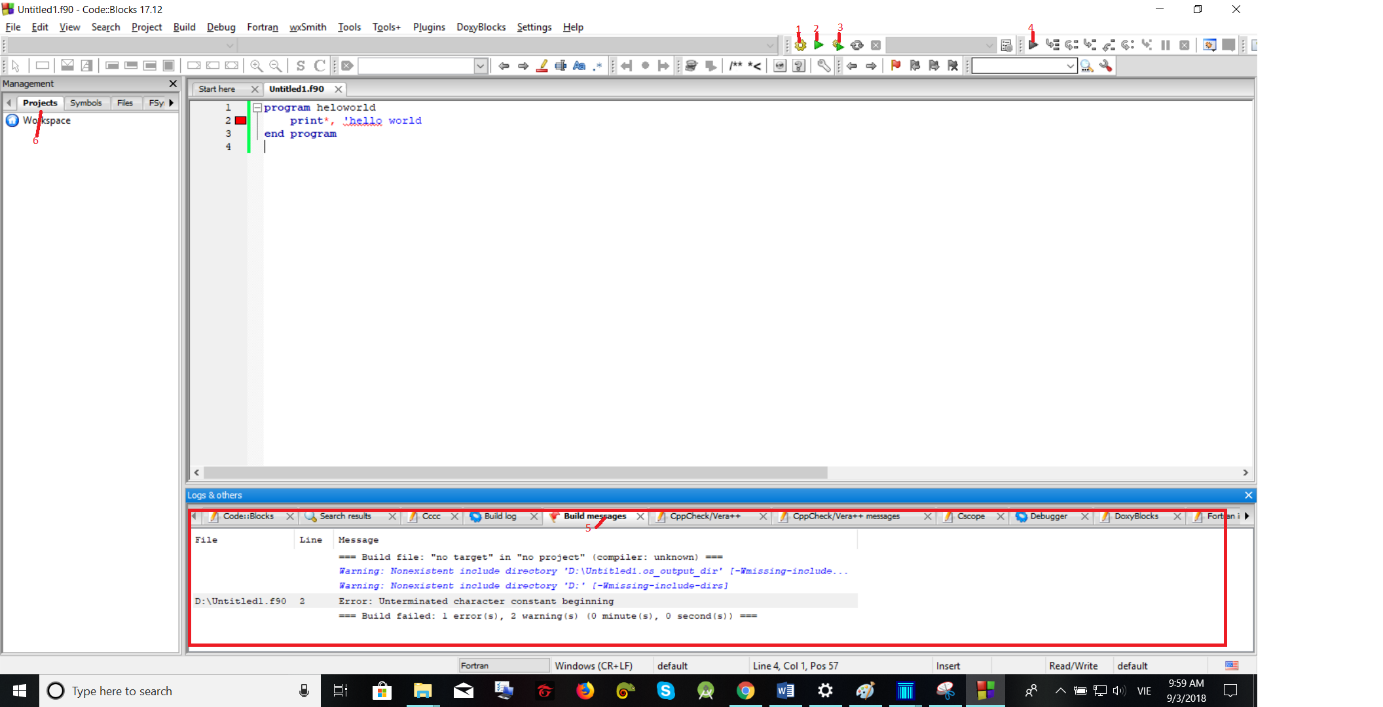
* **ấn ctrl + s lưa file dưới dạng tênfile.f90.**

**chương trình in ra dòng chữ hello world**





nhấn f9 hoặc nút buid and run ta được kết quả

các nút hay dùng trong codeblock

1.nút build: build trong file ra dạng file.exe

2. nút run dùng để chạy chương .exe trình đã build

3. nút build and run: build chương trình sau đó chạy

4. debug: dùng để chạy từng dòng lệnh gỡ lỗi chương trình

5. hộp hội thoại thông báo tình trạng chương trình, có lỗi hay không có build được hay lỗi

6. project tree: hiện thị cây project nếu creat project trong ví dụ trên chỉ tạo file nên không có

1. **Cấu trúc chung của một chương trình FORTRAN 90**
   * + 1. **Cấu trúc chương trình**

**[PROGRAM *TenChuongTrinh*]**

**[*Phan\_khai\_khai\_bao*]**

**[*Phan\_thuc\_thi*]**

**[CONTAINS  
*Các\_chương\_trình\_con\_trong* ]**

**END [PROGRAM [*TenChuongTrinh*]]**

Trong đó:

**+ [PROGRAM *TenChuongTrinh*]** :Đặt tên cho chương trình

**+ [*Phan\_khai\_khai\_bao*]**

Khai báo các biến, các thuộc tính, các hằng và kiểu dữ liệu tương ứng của chúng để trình biên dịch cấp phát bộ nhớ, phân luồng xử lý

+ **[*Phan\_thuc\_thi*]**

Là những câu lệnh xác định quy tắc và trình tự thực hiện tính toán, xử lý để đạt kết quả.

+ **[*Cac\_chuong\_trinh\_con*]**

Là các chương trình phụ dùng thực thi một số phép tính hay các thao tác xử lý dữ liệu nhiều lần. Nó làm cho chương trình gọn và dễ hiểu hơn.

+ **END [PROGRAM [*TenChuongTrinh*]]**

Câu lệnh này báo cho chương trình dịch rằng không còn câu lệnh nào nữa để dịch.

* + - 1. **Kiểu dữ liệu:**
* Fortran 90 định nghĩa 5 kiểu dữ liệu chuẩn, được chia thành hai lớp là lớp các kiểu số (numeric*)* gồm số nguyên (**integer**), số thực (**real**) và số phức (**complex**), và lớp các kiểu không phải số (non-numeric*)* gồm kiểu ký tự (**character**) và kiểu lôgic (**logical**).

|  |  |  |
| --- | --- | --- |
| **Cách khai báo** | **Số byte chiếm giữ** | **Phạm vi giá trị** |
| INTEGER | 4 | −2 147 483 648 đến 2 147 483 647 |
| INTEGER\*1 hoặc  INTEGER (1) hoặc INTEGER  (KIND=1) | 1 | −128 đến 127 |
| INTEGER\*2 hoặc  INTEGER (2) hoặc INTEGER  (KIND=2) | 2 | −32 768 đến 32 767 |
| INTEGER\*4 hoặc  INTEGER (4) hoặc INTEGER  (KIND=4) | 4 | −2 147 483 648 đến 2 147 483 647 |

* + - 1. **Các toán tử:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ký hiệu phép toán** | **Tên gọi/Ý nghĩa** | **Thứ tự ưu tiên** | **Thứ tự thực hiện** | **Ví dụ** |
| **Phép toán số học** | |  |  |  |
| \*\* | Phép lũy thừa | 1 | Phải sang trái | A \*\* B |
| \* | Phép nhân | 2 | Trái sang phải | A \* B |
| / | Phép chia | 2 | Trái sang phải | A / B |
| + | Phép cộng | 3 | Trái sang phải | A + B |
| − | Phép trừ | 3 | Trái sang phải | A − B |
|  |  |  |  |  |
|  | **Phép toán quan hệ** |  |  |  |
| .EQ. ( == ) | Bằng | − | Không phân định | A.EQ.B; A == B |
| .LT. ( < ) | Nhỏ hơn | − | Không phân định | A.LT.B; A < B |
| .LE. ( <= ) | Nhỏ hơn hoặc bằng | − | Không phân định | A.LE.B; A <= B |
| .GT. ( > ) | Lớn hơn | − | Không phân định | A.GT.B; A > B |
| .GE. ( >= ) | Lớn hơn hoặc bằng | − | Không phân định | A.GE.B; A >= B |
| .NE. ( /= ) | Không bằng (Khác) | − | Không phân định | A.NE.B; A /= B |
|  | **Phép toán lôgic** |  |  |  |
| .NOT. | Phủ định | 1 | Không phân định | .NOT. L1 |
| .AND. | Và (Phép hội) | 2 | Trái sang phải | L1. AND. L2 |
| .OR. | Hoặc (Phép tuyển) | 3 | Trái sang phải | L1. OR. L2 |
| .XOR. | Hoặc triệt tiêu | 4 | Trái sang phải | L1. XOR. L2 |
| .EQV. | Tương đương | 4 | Trái sang phải | L1. EQV. L2 |
| .NEQV. | Không tương đương | 4 | Trái sang phải | L1. NEQV. L2 |
|  | **Gộp ký tự** |  |  |  |
| // | Gộp hai xâu ký tự | − | Trái sang phải | ST1 // ST2 |

* + - 1. **Vòng lặp:**

**DO Biến\_Điều\_Khiển = GiaTriDau, GiaTriCuoi [, Buoc\_Nhảy]** Các\_câu\_lệnh **End Do**

* + - 1. **Lệnh IF**

**IF (BThuc\_Logic) THEN Các\_câu\_lệnh\_1**

**ELSE**

**Các\_câu\_lệnh\_2**

**END IF**

* + - 1. **Hàm (Function) , chương trình con**

Hàm trong có thể được khai báo như sau.  
**[*KiểuDL*][RECURSIVE] FUNCTION *TenHam*  
([*Các\_đối\_số*]) [RESULT (*TenKetQua*)]  
[*Các\_câu\_lệnh\_khai\_báo*]  
[*Các\_câu\_lệnh\_thực hiện*]  
[*TenHam* = ...]  
END FUNCTION [*TenHam*]**trong đó:   
***KiểuDL*** là kiểu dữ liệu mà hàm sẽ trả về. Ta có thể bỏ qua tùy chọn này khi sử dụng tùy  
chọn ***RESULT***.

**RECURSIVE** là tùy chọn để chỉ hàm là hàm đệ qui.  
***TenHam*** là tên của hàm, được dùng để gọi tới hàm.  
***Các\_đối\_số*** là danh sách các đối số hình thức, liệt kê cách nhau bởi dấu phẩy.  
***TenKetQua*** là tên biến chứa kết quả trả về của hàm. Nếu sử dụng tùy chọn này thì câu  
lệnh ***TenHam = ...*** không được phép xuất hiện. Ngược lại, nếu không sử dụng tùy chọn  
**RESULT** thì phải có dòng lệnh ***TenHam = ...*** để trả về kết quả của hàm.

**Ví dụ chương trình kiểm tra số chẵn lẻ:**

**program chanle**

**integer a**

**print\*, "nhap a: "**

**read\*, a**

**if (MOD(a,2).EQ.0) then**

**print\*, "so: ",a," la so chan"**

**else**

**print\*, "so: ",a," la so le"**

**end if**

**end program**

1. Ứng dụng ngôn ngữ Fortran

Ngôn ngữ Fortran khá tiện dụng trong nhiều lĩnh vực của thực tiễn, nhất là những bài toán phức tạp yêu cầu thời gian giải đáp thấp thì Fortran luôn đáp ứng tốt. Chính vì thế mà nó được áp dụng vào một số bài toán hiện nay như Dự đoán khí tượng thuỷ văn, ứng dụng trong ngành Công Nghệ Thông Tin, kỹ thuật lập trình phục vụ nhu cầu học tập.

VI Thuật toán QuickSort dưới dạng tuần tự

program MainQuickSort

implicit none

integer a(8)

a( 1) = 1

a( 2) = 1

a( 3) = 5

a( 4) = 3

a( 5) = -1

a( 6) = 11

a( 7) = 7

a( 8) = 31

print\*,"Mang dau vao la:"

print\*,a

print\*

call QuickSort(a,1,8)

print\*,"Mang da sap xep la:"

print\*,a

end program MainQuickSort

recursive subroutine QuickSort(a,first,last)

implicit none

integer a(8),mid,tg

integer first,last

integer i,j

mid = a((first+last)/2)

i = first;

j = last

do

do while (a(i)<mid)

i=i+1

end do

do while (mid<a(j))

j=j-1

end do

if (i>=j) exit

tg=a(i)

a(i)=a(j)

a(j)=tg

i=i+1

j=j-1

end do

if(first<i-1) call QuickSort(a,first,i-1)

if(j+1<last) call QuickSort(a,j+1,last)

end subroutine QuickSort