1. **CÀI ĐẶT MÔI TRƯỜNG LẬP TRÌNH C:**

Mọi source code ngồn ngữ C đều gồm 2 phần

* Khai báo thư viện **#include <tên thư viện>.**
* Phần hàm main – phần hàm chính của chương trình  **int main(){**

**// chương trình**

**}.**

* Ngôn ngữ C mặc định không làm được nhiều việc, bạn cần thêm thư viện để chúng làm được nhiều việc hơn, chẳng hạn như thư viện **stdio.h** giúp chúng ta có thể nhập xuất. Như trong bài “hello word”, hàm **printf()** dùng để xuất dữ liệu ra.
* Code: **D:\Files code\Hello.exe.**

1. **BIÊN DỊCH CODE:**

* gcc/g++ là một trình biên dịch code thành file chạy dược trên điều hành của bạn. VD như windowns thì sẽ biên dịch đuôi **.c; .cpp** thành **.exe**.
* Quá trình biên dịch cũng là quá trình phát hiện lỗi cú pháp. Nếu chương trình có lỗi, trình biên dịch sẽ dừng qúa trình biên dịch. File **.exe** chỉ được tạo ra khi biên dịch không lỗi.
* Kiểm tra phiên bản gcc: **gcc --version**
* Biên dịch code: **gcc file\_name.c(.cpp) –o after\_compiler\_name**
* Giúp đỡ: **gcc --help**
* Code: **D:\Files code\ttong.exe.**

1. **BIẾN VÀ HẰNG TRONG C:**

* Biến: là giá trị có thể thay đổi bất cứ lúc nào trong suốt quá trình mà chương trình chạy.
* Cách khai báo biến:

<kiểu dữ liệu> <tên biến>

Vd: int a;

char nho;

* Hằng: là một giá trị hằng số không cho phép thay đổi trong suốt quá trình chạy chương trình.
* Cách khai báo hằng:

const <kiểu dữ liệu> <tên hằng> = <giá trị>

Vd: const int a = 7;

const float pi = 3,14

* Code: **D:\Files code\bien\_hang.exe.**

1. **KIỂU DỮ LIỆU TRONG C:**

* Kiểu số nguyên:

|  |  |  |
| --- | --- | --- |
| **Type** | **Kích thước** | **Phạm vi giá trị** |
| char | 1 byte | -128 to 127 hoặc 0 to 255 |
| unsigned char | 1 byte | 0 tới 255 |
| signed char | 1 byte | -128 tới 127 |
| int | 2 or 4 bytes | -32,768 tới 32,767 hoặc -2,147,483,648 tới 2,147,483,647 |
| unsigned int | 2 or 4 bytes | 0 tới 65,535 hoặc 0 tới 4,294,967,295 |
| short | 2 bytes | -32,768 tới 32,767 |
| unsigned short | 2 bytes | 0 tới 65,535 |
| long | 8 bytes | -9223372036854775808 tới 9223372036854775807 |
| unsigned long | 8 bytes | 0 tới 18446744073709551615 |

* Kiểu số thực:

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Kích thước** | **Phạm vi giá trị** | **Độ chính xác** |
| float | 4 byte | 1.2E-38 to 3.4E+38 | 6 chữ số thập phân |
| double | 8 byte | 2.3E-308 to 1.7E+308 | 15 chữ số thập phân |
| long double | 10 byte | 3.4E-4932 to 1.1E+4932 | 19 chữ số thập phân |

* Kiểu kí tự:

|  |  |  |
| --- | --- | --- |
| **Type** | **Kích thước** | **Phạm vi giá trị** |
| char or signed char | 1 byte | -128 to 127 |
| unsigned char | 1 byte | 1. to 255 |

* Kiểu void: có nghĩa là “không có giá trị” , nó không được dùng trong khai báo biến thông thường mà được sử dụng để chỉ định kiểu của các hàm không giá trị trả về.
* Code: **D:\Files code\kieudl.exe.**

1. **HIỆN TƯỢNG TRÀN SỐ:**

* Hiện tượng tràn số là một lỗi cơ bản trong lập trình C, ta có thể tránh bằng cách sử dụng kiểu dữ liệu phù hợp.
* Code: **D:\Files code\transo.exe.**

1. **NHẬP XUẤT TRONG C:**

* Hàm printf():
* Được sử dụng để in ra các ký tự, các chuỗi, các giá trị , … và hiển thị lên màn hình console.
* Sử dụng %d để hiển thị kiểu dữ kiệu nguyên; %c để hiển thị kiểu kí tự; %f để hiển thị giá trị thực và %s để hiển thị kiểu dữ liệu chuỗi; %lf cho kiểu double và %x cho kiểu số ở hệ thập lục phân.
* Để xuống dòng ta sử dụng kí tự “\n.”

| **Kiểu dữ liệu** | **Định dạng** |
| --- | --- |
| Int | %d |
| Char | %c |
| Float | %f |
| Double | %lf |
| short int | %hd |
| unsigned int | %u |
| long int | %li |
| long long int | %lli |
| unsigned long int | %lu |
| unsigned long long int | %llu |
| signed char | %c |
| unsigned char | %c |
| long double | %Lf |

* Hàm scanf():
* Hàm được dử dụng để đọc vào các kí tự, chuỗi, các giá trị số mà người dùng nhập từ bàn phím.
* Hàm scanf() nhận vào tham số và địa chỉ của một biến đã được khai báo, xem VD: code của bài: **D:\Files code\hamscan.exe.**

#include <stdio.h> // standard input output

int main(){

    // Xuất printf()

    printf("Hello World!");

    // Nhập 1 số nguyên

    int a;

    printf("\nNhap a = ");

    scanf("%d", &a); // đừng quên dấu &

    // xuất có định dạng (format)

    printf("Gia tri a = %d", a); // Ctrl + /

    float b;

    printf("\nNhap b = ");

    scanf("%f", &b); // đừng quên dấu &

    // xuất có định dạng (format)

    printf("Gia tri b = %.2f", b);

    char c;

    printf("\nNhap c = ");

    scanf("%d", &c); // đừng quên dấu &

    // xuất có định dạng (format)

    printf("Gia tri c = %c", c);

    // Nhập nhiều biến đồng thời

    int a1, a2;

    printf("\nHay nhap a1 va a2: ");

    scanf("%d%d", &a1, &a2);

    // Xuat format

    printf("\nGia tri \"a1\" = %d\nGia tri \"a2\" = %d", a, a2);

    // In ra tiếng kêu bíp

    printf("a\bd");

}

* Kết quả:

Hello World!

Nhap a = 13

Gia tri a = 13

Nhap b = 3.1456

Gia tri b = 3.15

Nhap c = e

Gia tri c =

Hay nhap a1 va a2: 23 45

Gia tri "a1" = 13

Gia tri "a2" = 45d

1. **TOÁN TỬ TRONG C:**

* Toán tử số học: được dùng để thực hiện các phép toán: cộng, trừ, nhân, chia,… trên các giá trị số. Đây là toán tử cần có 2 giá trị số nên được phân loại là tón tử 2 ngôi.

| **Toán tử** | **Ý nghĩa** |
| --- | --- |
| + | phép toán cộng |
| – | phép toán trừ |
| \* | phép toán nhân |
| / | phép toán chia |
| % | phép toán lấy số dư(chỉ áp dụng cho số nguyên) |

* Code: **D:\Files code\ttuso.exe**.
* Toán tử tăng giảm: là toán tử một ngôi
* Toán tử ++: Tăng giá trị lên 1 đơn vị
* Toán tử --: Giảm giá trị đi 1 đơn vị
* Code: **D:\Files code\tttg.exe**.
* Toán tử gán: được dùng để gán giá trị cho 1 biến trong ngôn ngữ lập trình C

| **Toán tử** | **Viết gọn** | **Viết đầy đủ** |
| --- | --- | --- |
| = | a = b | a = b |
| += | a += b | a = a+b |
| -= | a -= b | a = a-b |
| \*= | a \*= b | a = a\*b |
| /= | a /= b | a = a/b |
| %= | a %= b | a = a%b |

* Code: **D:\Files code\ttgan.exe.**
* Toán tử quan hệ: được dùng để kiểm tra mối quan hệ giữa 2 toán hạng. Nếu quan hệ kiểm tra là đúng thì nó trả về giá trị TRUE và trả về FALSE trong trường hợp ngược lại.

| **Toán tử** | **Ý nghĩa** | **Ví dụ** |
| --- | --- | --- |
| == | so sánh bằng | 7 == 3 cho kết quả là 0 |
| > | so sánh lớn hơn | 5 > 1 cho kết quả là 1 |
| < | so sánh nhỏ hơn | 5 < 2 cho kết quả là 0 |
| != | so sánh khác | 5 != 4 cho kết quả là 1 |
| >= | lớn hơn hoặc bằng | 8 >= 3 cho kết quả là 1 |
| <= | nhỏ hơn hoặc bằng | 5 <= 0 cho kết quả là 0 |

* Code: **D:\Files code\quanhe.exe.**
* Toán tử logic:
* Toán tử &&: là toán tử AND, trả về true khi và chỉ khi tất cả các toán hạng đều đúng.
* Toán tử ||: là toán tử OR, trả về true khi có ít nhất 1 toán hạng đúng.
* Toán tử !: là toán tử NOT, phủ định giá trị của toán hạng.
* Code: **D:\Files code\logic.exe.**
* Các toán tử khác:
* Toán tử bit.
* Toán tử phẩy.
* Toán tử sizeof

1. **IN MÃ ASCII CỦA KÍ TỰ NHẬP TỪ BÀN PHÍM:**

* Code: **D:\Files code\maascii.exe.**

1. **HOÁN VỊ HAI SỐ:**

* Hoán vị hai số sử dụng biến tạm: **D:\Files code\bientam.c**

Code: **D:\Files code\bientam.exe**

* Hoán vị hai số sử dụng + và -: **D:\Files code\hvbdau.c**

Code: **D:\Files code\hvbdau.exe**

* Hoán vị dùng \* và /: **D:\Files code\hvnhanchia.c**

Code: **D:\Files code\hvnhanchia.exe**

* Hoán vị dùng XOR: **D:\Files code\hvxor.c**

Code: **D:\Files code\hvxor.exe**

1. **CÂU LỆNH IF ELSE:**

* Câu lệnh if:
* Cấu trúc: if () {

// khối lệnh sẽ thực hiện

}

* Code: **D:\Files code\if.exe**
* Câu lệnh if else:
* Cấu trúc: if () {

// khối lệnh

} else {

// khối kệnh

}

* Code: **D:\Files code\ifelse.exe**
* Câu lệnh if … else if … else
* Cấu trúc: if () {

} else if () {

} else if () {

}

* Code: **D:\Files code\noitiep.exe**
* Câu lệnh if else lồng nhau:
* If () {

If () {

}

}

* Code: **D:\ Files code\long.exe**

1. **CHƯƠNG TRÌNH XẾP LOẠI HỌC SINH:**

Code: **D:\ Files code\xeploai.c**

**D:\ Files code\xeploai.exe**

1. **TÌM SỐ LỚN NHẤT TRONG 3 SỐ a, b, c:**

Code: : **D:\ Files code\timmax.c**

**D:\ Files code\timmax.exe**

1. **VÒNG LẶP TRONG C:**

* Vòng lặp for:
* Cú pháp: **for (khởi tạo giá trị biến lặp; điều kiện lặp; cặp nhật biến lặp){**

**// lệnh cần lặp**

**}**

* Code 1:

#include <stdio.h>

int main(){

for (int i = 0; i < 3; i ++){

printf("lap 3 lan\n");

}

printf("ket thuc");

}

* Code 2:

#include <stdio.h>

int main(){

int a =5;

int b= 32

for (int i = a; i < b; i++){

printf(“\t%d”, i);

}

}

* Code 3:

#include <stdio.h>

int main(){

int a ;

int b ;

scanf("%d", &a);

scanf("%d", &b);

int sum = 0; // cần khởi tạo giá trị

for(int i = a; i <= b; i++){

sum += i; // sum = sum + i

}

printf("Sum = %d", sum);

}

1. **VÒNG LẶP WHILE VÀ DO WHILE:**

* Vòng lặp while thường dùng cho vòng lặp không biết trước số lần lặp:
* Cú pháp: while (testExpression){

// statements inside the body of the loop

}

* Code minh họa: **D:\ Files code\lapwhile.c**

**D:\ Files code\lapwhile.exe**

* Vòng lặp do …while: khác biệt đó là nó sẽ chạy phần thân trước 1 lần dù điều kiện đúng hay sai.
* Cú pháp: do{

//

While(testExpression);

* Code minh họa: **D:\ Files code\lapdowhile.c**

**D:\ Files code\lapdowhile.exe**