1. Kiểu dữ liệu Bool là kiểu dữ liệu có 2 giá trị TRUE ( Đúng ); FALSE ( Sai ). Thường để kiểm tra logic trong phép tính hoặc 1 kết quả so sánh.

Cách khai báo

Bool isTrue=True;

Bool isFalse=False;

1. Toán tử 3 ngôi trong C:

Variable=(Condition Expression) ? expression 1 : expression2;

Khi condition là True sẽ gán giá trị của expression 1 vào Variable. Khi condition là False sẽ gán giá trị của expression 2 vào Variable.

Vdu:

Int x=3;

Int y;

Y = ( x % 2 == 1 ) ? 1 : 0;

Cout << y;

Ở đây x số dư của x = 1 sẽ là True, số dư là 0 là False nên khi x = 3 chia 2 sẽ dư 1 là đúng và gán 1 vào y. Nếu gán x = 4 thì y sẽ = 0 vì biển thức điều kiện là dư 1 nên dư 0 là sai, vậy giá trị 0 sẽ được gán cho y.Như vậy phép toán tử 3 ngôi sẽ ngắn hơn if else trong các bài toán đơn giản.

1. CƠ CHẾ LƯU TRỮ BIẾN TRONG LẬP TRÌNH C
   1. - Khi một biến được tạo ra sẽ gắn với một địa chỉ bộ nhớ( vị trí lưu biến). Khi gắn giá trị với biến thì nó sẽ nằm trong địa chỉ bộ nhớ( để truy cập nó thì dùng toán tử tham chiếu(&)).
   2. - Nhiều loại biến khác nhau=> Các lớp lưu trữ khác nhau. Lớp lưu trữ xác định phạm vi và thời gian tồn tại của các biến. Bốn lớp lưu trữ C, auto, register, extern, static.
   3. + auto: lớp lưu trữ mặc định cho các biến trong hàm, chỉ có thể được truy cập bởi hàm chứa chúng và bị hủy khi hàm kết thúc, 'auto' chỉ có thể được sử dụng trong các hàm, tức là các biến cục bộ.
   4. +register: lớp lưu trữ dùng để xác định các biến cục bộ cần được lưu trữ trong thanh ghi thay vì RAM=> biến có kích thước tối đa bằng kích thước thanh ghi (thường là một từ) và không thể áp dụng toán tử ‘&’ cho nó (vì nó không có vị trí bộ nhớ). Các biến này có thể được truy cập nhanh hơn các biến auto, nhưng số lượng của chúng bị giới hạn bởi số lượng thanh ghi có sẵn. Thanh ghi chỉ được sử dụng cho các biến yêu cầu truy cập nhanh như bộ đếm. Cũng cần lưu ý rằng việc xác định 'register' không có nghĩa là biến sẽ được lưu trong sổ register mà là nó có thể được lưu trữ trong một thanh ghi nào đó tùy thuộc vào phần cứng và các hạn chế thực hiện.

+static: Lớp lưu trữ static chỉ thị trình biên dịch để giữ một biến cục bộ tồn tại trong suốt thời gian chạy của chương trình thay vì tạo và hủy nó mỗi lần nó đi vào và đi ra khỏi phạm vi.

Lớp lưu trữ Static cũng có thể được áp dụng cho các biến toàn cục. Khi điều này được thực hiện, nó làm cho phạm vi của biến đó bị hạn chế đối với tệp mà nó được khai báo.

Trong lập trình C, khi static được sử dụng trên các lớp, nó chỉ gây ra một bản sao của thành viên đó được chia sẻ bởi tất cả các đối tượng của lớp của nó.

* 1. + extern: Đây là lớp lưu trữ được sử dụng để cung cấp một tham chiếu của một biến toàn cục hiển thị cho TẤT CẢ các tệp chương trình. Khi bạn sử dụng ‘extern’, biến không thể được khởi tạo vì tất cả những gì nó làm là trỏ tên biến vào vị trí lưu trữ của biến được khai báo ở một nơi khác3. Các biến này có thể được truy cập bởi bất kỳ hàm nào trong chương trình.

1. TẠI SAO ĐANG NHẬP SỐ MÀ CHUYỂN SANG NHẬP KÍ TỰ THÌ BỊ BỎ QUA KHÔNG NHẬP ĐƯỢC VÀ TẠI SAO DÙNG FFLUSH STDIN?

-Đây là một vấn đề thường gặp khi sử dụng hàm scanf để nhập số và ký tự trong C. Nguyên nhân là do hàm scanf chỉ đọc dữ liệu từ bàn phím cho đến khi gặp ký tự khoảng trắng, dấu chấm phẩy, hoặc dấu xuống dòng. Khi nhập số, hàm scanf sẽ để lại dấu xuống dòng trong bộ nhớ đệm, và khi nhập ký tự, hàm scanf sẽ đọc dấu xuống dòng đó và bỏ qua việc nhập ký tự.

-Cách khắc phục là dùng hàm getchar để đọc và xóa dấu xuống dòng trong bộ nhớ đệm trước khi nhập ký tự

- hàm fflush stdin dùng để xóa bộ nhớ đệm, tránh dữ liệu cũ trong bộ nhớ đệm ảnh hưởng đến dữ liệu mới.