1.CƠ CHẾ LƯU TRỮ BIẾN TRONG LẬP TRÌNH C

- 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(&)).

- 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.

+ 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.

+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.

+static: lớp lưu trữ dùng để xác định các biến có thời gian tồn tại trong suốt chương trình, nhưng chỉ có thể được truy cập trong phạm vi được xác định. Các biến này được khởi tạo một lần và giữ giá trị của chúng ngay cả khi hàm chứa chúng kết thúc. Các biến này cũng có thể được khai báo ở cấp độ tệp, để tạo ra các biến toàn cục chỉ có thể được truy cập bởi các hàm trong cùng một tệp.

+ 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.

2. TÌM HIỂU VỀ KIỂU DỮ LIỆU BOOLEAN VÀ CÁCH DÙNG TRONG C

- Bool là kiểu dữ liệu chỉ nhận giá trị true or false, trong C thì dùng integer biểu diễn true/false( 0 thì T, khác 0 thì F), bool được dùng để biểu diễn gt logic, so sánh, điều kiện.

- cách dùng:

• Sử dụng thư viện <stdbool.h>

• Tự định nghĩa kiểu bool với enum

• Tự định nghĩa kiểu bool vơi integer

• Khai báo hằng số với true/false.

3. TÌM HIỂU VỀ TOÁN TỬ BA NGÔI TRONG C, LẤY VÍ DỤ, DEMO.

-Là toán tử có ba toán hạng, dùng để thực hiện phép toán có điều kiện

Cú pháp condition ? true\_value : false\_value

+ condition là kiểm tra điều kiện true or false

+ true\_value là gt sẽ đc thực hiện nếu true

+ False\_value ngược lại

- toán từ ba ngôi dùng để rút gọn lệnh if... else trong trường hợp đơn giản.

Ví dụ:

#include <iostream>

using namespace std;

int main() {

int number = 10;

string result;

if (number % 2 == 0) {

result = "Số chẵn";

} else {

result = "Số lẻ";

}

cout << result << endl;

return 0;

} (trước khi dùng)

=> sau khi dùng

#include <iostream>

using namespace std;

int main() {

int number = 10;

string result = (number % 2 == 0) ? "Số chẵn" : "Số lẻ";

cout << result << endl;

return 0;

}

4. 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.