1. **Bản chất biến trong một chương trình C là gì. Phân biệt biến toàn cục, biến cục bộ và biến môi trường. Cho ví dụ minh họa.**

**- Bản chất của Biến**: là vùng nhớ dùng để lưu trữ các giá trị của chương trình. Mỗi biến gắn liền với một kiểu dữ liệu và một định danh duy nhất gọi là tên biến.

- Biến toàn cục: Biến được khai báo bên ngoài phạm vi các hàm, biến này được dùng chung cho các hàm(dùng cho cả chương trình)

- Biến cục bộ: là biến được dùng và khai báo trong 1 hàm và chỉ được dùng trong hàm đã khai báo nó.

- Biến môi trường: là môi trường để chạy một ngôn ngữ lập trình. Ví dụ để chạy được java thì cần có JDK

1. **Phân biệt giữa biên dịch và thông dịch; cho ví dụ minh họa. Mô tả quá trình biên dịch một chương trình C; chỉ rõ các file trung gian tạo ra khi biên dịch trong IDE CodeBlock**

Biên dịch: chuyển các câu lệnh được gõ bằng 1 ngôn ngữ lập trình(mã nguồn) sang một chương trình tương đương nhưng bằng một ngôn ngữ máy tính mới (chương trình đích). Lần sau muốn chạy lại chương trình, chỉ cần chạy lại chương trình đã được dịch.  
Ví dụ: khi soạn xong 1 chương trình C, sau khi biên dịch bạn sẽ được chương trình dạng mã máy (.exe)

Thông dịch: sau khi soạn thảo một chương trình thì quá trình thông dịch là quá trình xảy ra lúc runtime (khi thực hiện), trình thông dịch sẽ dịch từng lệnh của chương trình và thực thi. Lần sau muốn chạy lại chương trình thì phải thông dịch lại.

Ví dụ: php hoặc asp. Mỗi khi bạn chạy website, trình thông dịch sẽ dịch lại từ đầu mã nguồn và thực thi.

File trung gian tạo ra khi biên dịch trong IDE CodeBlock:

-File.exe:

-File.o

1. **Mô tả quá trình cấp phát bộ nhớ trong quá trình thực hiện một chương trình C.**

Khi ta viết 1 chương trình hay khai báo 1 biến trong 1 chương trình thì chúng đều có giá trị và địa chỉ của chúng. Khi chương trình khởi chạy, hệ điều hành sẽ phân ra thành 4 vùng nhớ trong RAM. Biến và mã nguồn sẽ được lưu trữ tại các vùng nhớ này. Các vùng nhớ:

Code Segment: Mã nguồn của chúng ta sau khi hoàn tất quá trình Build sẽ chuyển thành các đoạn mã máy 0 và 1. Khi chương trình khởi chạy, các đoạn mã máy này sẽ được nạp vào Code Segment. Cách duy nhất để truy xuất vào vùng nhớ này là sử dụng Con trỏ hàm. Code Segmentcó kích thước cố định

Data Segment: Khi chương trình được khởi chạy, toàn bộ các biến toàn cục và static đều được lưu trữ ở đây, các đoạn chuỗi cố định cũng được lưu trữ trong Data Segment. Data Segment có kích thước cố định.

Stack Segment: Đây là vùng nhớ mà chúng ta cần quan tâm. Khi một hàm được gọi, hàm đó sẽ được đưa vào vùng nhớ Stack, các biến được khai báo trong hàm đó cũng được đưa vào vùng nhớ Stack. Khi hàm kết thúc, toàn bộ các biến trong hàm cùng với bản thân hàm sẽ được tự động giải phóng để các hàm sau sử dụng. Stack Segment có kích thước cố định

Heap Segment: Đây là vùng nhớ khác mà chúng ta cần quan tâm, vùng nhớ này chúng ta phải hoàn toàn kiểm soát nó. Khi chúng ta sử dụng con trỏ và cấp phát động một vùng nhớ cho con trỏ quản lý, vùng nhớ này sẽ nằm trong Heap Segment. Heap Segment có kích thước không cố định

1. **Mô tả quá trình thực hiện và tổ chức các thành phần bên trong bộ nhớ của một hàm đệ quy.**

**-** Khi gọi đệ quy, trình biên dịch sẽ lưu tất cả các giá trị trong hàm tính toán được và những việc cần làm vào 1 stack ví dụ như ta có đoạn chương trình sau với n = 3;

int sum(int a){

if (n == 0) return 0;

return (n + sum(n – 1));

}

- Trình biên dịch sẽ đưa lần lượt các việc cần thực hiện trong hàm vào stack, điển hình đầu tiên n sẽ được đưa vào stack, cụ thể ở đây là 3 và sau đó nó lại bắt đầu gọi về sum. Lúc này sẽ trỏ về sum và stack tăng lên 1 đồng thời lưu trữ giá trị tiếp theo là 2. Tương tự vậy cho 1 và đến 0 thực hiện hàm return về 0 lúc này các giá trị sẽ được lần lượt lấy ra khỏi stack và trả về địa chỉ hàm main. Các giá trị được lấy ra như sau: 0 + 1 + 2 + 3 = 6 và sum nhận giá trị 6.

1. **Bản chất quá trình chuyển điều khiển giữa chương trình chính và chương trình con. Tại sao trong chương trình sử dụng nhiều chương trình con lại làm tăng thời gian thực hiện.**

- Khi hàm main bắt gặp việc gọi một chương trình con, ngay lúc này trình biên dịch sẽ khởi tạo 1 stack để với giá trị ban đầu là địa chỉ của câu lệnh tiếp theo sẽ được thực hiện ở hàm main và các giá trị của chương trình con cũng được thực hiện ở stack đó. Ví dụ như sau:

void In(){

printf(“Hello”);

}

int main(){

int a, b, c;

In();

printf(“finish”);

return 0;

}

Phân tích đoạn chương trình sau ta sẽ thấy được như sau: Đầu tiên ta sẽ thực hiện hàm main với các giá trị được khởi tạo là a, b và c. Sau đó, trình biên dịch bắt gặp việc gọi chương trình con thì ngay lập tức địa chỉ câu lệnh tiếp theo sẽ được đưa vào stack và bắt đầu thực hiện void In() - printf(“Hello”) sẽ được thực hiện và cuối cùng địa chỉ câu lệnh tiếp theo của hàm main sẽ được lấy ra khỏi stack và tiếp tục thực hiện chương trình.

1. **Giải thích cách tính miền giá trị của các kiểu dữ liệu. Mô tả cách tổ chức của mảng trong bộ nhớ. Trong C một mảng có tối đa bao nhiêu phần tử.**

Cách tính miền giá trị: ta có kiểu dữ liệu cần tính có kích thước n byte. Tìm số bit của kiểu dữ liệu đó bằng các lấy số n byte nhân cho 8.

Để tìm được miền giá trị ta lấy 2 lũy thừa cho số bit vừa tìm được và trừ đi 1 sẽ ra số nguyên x. Miền giá trị của kiểu dữ liệu đó từ -x -> x.

Mảng là một danh sách các ô nhớ nằm sát nhau, mỗi ô nhớ chứa một địa chỉ nhất định. Các ô nhớ trong mang có cùng kiểu dữ liệu với nhau.

Số phần tử tối đa trong mảng là 1 triệu nếu là mảng tính, 1 tỉ nếu là mảng động.

1. **Bản chất biến là gì; địa chỉ ô nhớ là gì; con trỏ là gì; con trỏ hàm là gì. Ý nghĩa sử dụng của con trỏ hàm; cho ví dụ minh họa.**

**Bản chất của Biến**: là vùng nhớ dùng để lưu trữ các giá trị của chương trình. Mỗi biến gắn liền với một kiểu dữ liệu và một định danh duy nhất gọi là tên biến.

Con trỏ đơn giản là một biến kiểu số nguyên dùng để chứa địa chỉ.

1. **Trong một chương trình C, bản chất biến là gì; địa chỉ ô nhớ là gì. Mô tả quá trình thực hiện của một chương trình.**

Biến là ô nhớ hay vùng nhớ được khai báo để lưu trữ một giá trị chỉ định trong chương trình

Một ô nhớ sẽ được xác định bởi một địa chỉ ô nhớ xác định

Quá trình thực hiện một chương trình:

Đầu tiên trình biên dịch sẽ khởi tạo 1 vùng nhớ để bắt đầu thực hiện chương trình. Và các cấu trúc lệnh sẽ được thực hiện một cách lần lượt và có thứ tự. Với mỗi cấu trúc lệnh sẽ có 1 phân vùng nhớ để thực hiện và trả giá trị về hàm main.