|  |  |
| --- | --- |
| **Con trỏ ( Pointer )** | một biến đặc biệt lưu trữ địa chỉ của một biến khác trong bộ nhớ. Các con trỏ thường được sử dụng để tham chiếu đến một vị trí cụ thể trong bộ nhớ, giúp chương trình truy cập và thao tác dữ liệu một cách linh hoạt hơn.  Cú pháp : dataType \*name;    Lưu ý: biến con trỏ kiểu int\* trong C cũng có thể lưu trữ địa chỉ của biến kiểu char. Tuy nhiên, khi sử dụng con trỏ kiểu int\* để truy cập địa chỉ của một biến kiểu char, có thể gây ra các vấn đề về độ dài của kiểu dữ liệu và tính toàn vẹn của dữ liệu. |
| **Các cách tạo mảng động trong C** | 1. Sử dụng hàm malloc():   - Hàm malloc() được sử dụng để cấp phát bộ nhớ động trong C.Khi giải phóng bộ nhớ bằng cách sử dụng hàm free() thì các phần tử trong mảng vẫn tồn tại và giá trị của chúng không thay đổi.  - Tuy nhiên, sau khi giải phóng bộ nhớ bằng hàm free(), mảng đó không còn được sử dụng nữa. Các phần tử trong mảng có thể bị thay đổi giá trị nếu vùng bộ nhớ trước đó đã được cấp phát cho một biến khác. Do đó, để tránh việc mất dữ liệu trong mảng, bạn nên đảm bảo không sử dụng mảng đó sau khi giải phóng bộ nhớ.  - Lưu ý rằng sau khi sử dụng mảng động, bạn cần giải phóng bộ nhớ của nó bằng cách sử dụng hàm free() để tránh lãng phí bộ nhớ.     1. Sử dụng hàm calloc():   - Hàm calloc() cũng được sử dụng để cấp phát bộ nhớ động trong C, tuy nhiên khác với malloc() là nó sẽ tự động khởi tạo các giá trị của mảng động bằng 0. |
| **Gets()** | Trong ngôn ngữ lập trình C, hàm gets() được sử dụng để đọc một chuỗi ký tự từ bàn phím và lưu trữ nó trong một mảng ký tự. |
|  |  |
| **Thư viện <String.h>** | - Thư viện string.h (hay còn được gọi là cstring) chứa các hàm và các hằng số để xử lý các chuỗi ký tự  Dưới đây là một số hàm quan trọng trong thư viện string.h:  **- strcpy(char \*dest, const char \*src)**: Sao chép chuỗi src vào chuỗi dest.  **- strcat(char \*dest, const char \*src)**: Nối chuỗi src vào cuối chuỗi dest.  **- strlen(const char \*str)**: Trả về độ dài của chuỗi str.  **- strcmp(const char \*str1, const char \*str2)**: So sánh hai chuỗi str1 và str2.  **- strchr(const char \*str, int c)**: Tìm kiếm ký tự c trong chuỗi str.  **- strstr(const char \*str1, const char \*str2)**: Tìm kiếm chuỗi str2 trong chuỗi str1. |