|  |  |
| --- | --- |
| **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. |
| **Null trong c** | Trong c null ta phải ghi in hoa lên toàn bộ là NULL |
| **Các cách tạo mảng động trong C** | - Nằm trong thư viện **stdlib.h**   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()** | - Nằm trong thư viện string.h  Trong ngôn ngữ lập trình C, hàm gets() có thể lưu được mọi ký tự đặc biệt kể cả dấu space    Hàm gets() không kiểm tra độ dài chuỗi nhập vào và có thể dẫn đến tràn bộ đệm và lỗi bảo mật nghiêm trọng. Do đó, từ phiên bản C11 trở đi, hàm gets() đã bị loại bỏ khỏi thư viện chuẩn C.  Gets không nhận kí tự enter |
| **Scanf()** | Trong ngôn ngữ lập trình C, hàm **scanf()** chỉ có thể lưu được các kiểu dữ liệu khác như int, float, char … nhưng không lưu được ký tự đặc biệt như hàm gets(). ví dụ như nó không lưu được dấu space. |
| **Fgets()** | Hàm fgets() trong ngôn ngữ lập trình C được sử dụng để đọc một chuỗi ký tự từ một file hoặc từ bàn phím (stdin) và lưu vào một mảng ký tự được chỉ định. Cú pháp của hàm fgets() như sau:    Hàm fgets() sẽ đọc chuỗi từ stream cho đến khi đọc được num - 1 ký tự hoặc gặp ký tự kết thúc dòng \n hoặc gặp ký tự kết thúc tập tin EOF. Sau khi đọc xong, hàm fgets() sẽ thêm ký tự kết thúc chuỗi \0 vào cuối chuỗi đọc được và trả về con trỏ đến chuỗi vừa đọc được.    Hàm fgets() yêu cầu truyền vào tham số num để xác định số lượng ký tự tối đa cần đọc.  Hàm fgets() có tính năng bảo mật hơn so với hàm gets() bởi vì nó chỉ đọc tối đa num ký tự và không cho phép đọc quá số ký tự được cấp phát. Điều này giúp ngăn chặn nguy cơ tràn bộ đệm và tấn công từ chối dịch vụ (DoS).  Fgets nhận luôn kí tự enter. |
| **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.  + Hàm này sẽ so sánh str1với chuỗi str2và trả về giá trị:  0 nếu hai chuỗi bằng nhau.  Một số âm nếu str1nhỏ hơn str2.  Một số dương nếu str1lớn hơn 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. |
| **Đọc ghi file** | |  |  | | --- | --- | | fopen() | mở tệp mới hoặc file đang tồn tại | | fprintf() | ghi dữ liệu vào file | | fscanf() | đọc dữ liệu từ file | | fputc() | ghi một ký tự vào file | | fgetc() | đọc một ký tự từ file | | fclose() | đóng file | | fseek() | đặt con trỏ tập tin vào vị trí đã cho | | fputw() | ghi một số nguyên vào file | | fgetw() | đọc một số nguyên từ file | | ftell() | trả về vị trí hiện tại | | rewind() | đặt con trỏ tập tin vào đầu tập tin | |
| **Fopen()** | hàm fopen được sử dụng để mở một tệp tin và trả về một con trỏ FILE để thao tác với tệp tin đó      Trong đó các chế độ mode   |  |  | | --- | --- | | r | mở file văn bản ở chế độ đọc | | w | mở file văn bản ở chế độ ghi | | a | mở file văn bản ở chế độ nối thêm nội dung vào nội dung sẵn có trong file. | | r+ | mở file văn bản ở chế độ đọc và ghi | | w+ | mở file văn bản ở chế độ đọc và ghi | | a+ | mở file văn bản ở chế độ đọc và ghi | | rb | mở file văn bản ở chế độ đọc và ghi | | wb | mở file nhị phân ở chế độ đọc | | ab | mở file nhị phân ở chế độ nối thêm nội dung vào nội dung sẵn có trong file. | | rb+ | mở file nhị phân ở chế độ đọc và ghi | | wb+ | mở file nhị phân ở chế độ đọc và ghi | | ab+ | mở file nhị phân ở chế độ đọc và ghi | |
| **Fprintf()** | hàm fprintf được sử dụng để ghi dữ liệu vào tệp tin theo định dạng đã cho    Trong đó:  - stream là con trỏ đến tệp tin cần ghi dữ liệu vào.  - format là chuỗi định dạng để ghi dữ liệu vào tệp tin.  - các đối số còn lại là các giá trị cần ghi vào tệp tin theo định dạng đã cho trong chuỗi format.  Hàm fprintf sẽ trả về số lượng ký tự đã ghi vào tệp tin, hoặc giá trị âm nếu có lỗi xảy ra. |
| **Fcanf()** | hàm fscanf được sử dụng để đọc dữ liệu từ tệp tin theo định dạng đã cho    Trong đó:   * stream là con trỏ đến tệp tin cần đọc dữ liệu từ. * format là chuỗi định dạng để đọc dữ liệu từ tệp tin. * các đối số còn lại là các con trỏ đến biến cần lưu giá trị được đọc từ tệp tin theo định dạng đã cho trong chuỗi format.   Trong đó:   * stream là con trỏ đến tệp tin cần đọc dữ liệu từ. * format là chuỗi định dạng để đọc dữ liệu từ tệp tin. * các đối số còn lại là các con trỏ đến biến cần lưu giá trị được đọc từ tệp tin theo định dạng đã cho trong chuỗi format. |