1. Cơ bản trong c
2. Trong lập trình, a++ và ++a đều tăng giá trị của biến a lên 1 đơn vị, NHƯNG:

* A++: tăng sau, đầu tiên giá trị của a đc sử dụng trong biểu thức, sau đó tăng thêm 1.
* ++a: đc tăng lên 1, sau đó mới được sử dụng trong biểu thức.

A screenshot of a computer

Description automatically generatedđịnh dạng và phạm vi của các kiểu dữ liệu trong c

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedII. cấu trúc rẽ nhánh

1. if(điều kiện){ câu lệnh}

else if(điều kiện){câu lệnh}

else{câu lệnh}

1. switch (x){

case giá trị 1:

câu lệnh

break;

case giá trị 2:

câu lệnh

break;

case giá trị 3:

câu lệnh

break;

default: //nếu không có giá trị nào thỏa mãn các trường hợp trên thì sẽ chạy cái này

câu lệnh

break;}

//khi ko cho lệnh break vào thì sẽ chạy từ trường hợp đến hết.

//nếu giá trị của x bằng giá trị 1 thì sẽ chạy case 1

có thể thêm ! vào với mục đích phủ đinh điều kiện

ví dụ if !(điều kiện){câu lệnh}

1. comment : /\*comment nhiều dòng\*/
2. vòng lặp
3. for(giá trị ban đầu;điều kiện;giá trị tăng giảm bao nhiêu){câu lệnh}

* hàm sẽ chạy giá trị 1 ròi kiểm tra điều kiện rồi chạy câu lệnh, chạy câu lệnh sẽ thực hiện tăng giảm giá trị đã gán.

1. Các câu lệnh điều khiển vòng lặp

* Break: là câu lệnh dùng để dừng vòng lặp.
* Continue: khi dùng câu lệnh này các đoạn code bên dưới sẽ không được thực thi

1. Do – while

* Do{

//khối lệnh

} while (điều kiện);

* Do while sẽ chạy câu lệnh trước rồi mới kiểm tra điều kiện vì thế vòng lập sẽ chạy ít nhất 1 lần. trong khi đó while sẽ kiểm tra điều kiện trước rồi mới chạy khối lệnh

1. Mảng
2. Mảng đưcọ hiểu là một tập hợp các biến. thay vì nhập một loạt biến thù có thể tạo mảng để dòng code được rút gọn và dễ nhìn hơn.

* Các phần tử của mảng được truy xuất bằng toán tử []. Ví dụ:
* A[0] tương đương với phần tử thứ nhất
* A[1] tương đương với phần tử thứ hai
* …
* Nhìn chung cách làm bài này với mảng cũng giống việc tạo ra nhiều biến chỉ khác ở chỗ thay vì phải viết a0, a1, a2,..,a9 thì bạn có thể dùng vòng lặp for và viết a[i]
* Ví dụ:
* #include<stdio.h>
* int main() {
* // Khai báo mảng a kiểu số nguyên với 10 phần tử
* int a[10];//a[10]={. . . };
* // Khai báo biến lưu trữ tổng các phần tử trong mảng
* int sum = 0;
* // Dùng vòng for để nhập dữ liệu cho các phần tử trong mảng
* for (int i = 0; i < 10; i++) {
* scanf("%d", &a[i]);
* }
* // Dùng vòng for để tính tổng các phần tử trong mảng
* for (int i = 0; i < 10; i++) {
* sum += a[i];
* }
* // In ra tổng các phần tử trong mảng
* printf("%d", sum);
* return 0;
* }
* Ngoài ra bạn có thể khai báo và khởi tạo giá trị cho mảng bằng 1 số cách dưới đây:

// Khai báo mảng a 5 phần tử với giá trị từ 1 tới 5

int a[5] = {1, 2, 3, 4, 5};

// Khai báo mảng a 5 phần tử với giá trị từ 1 tới 5

int a[] = { 1, 2, 3, 4, 5 };

// Khai báo mảng a 20 phần tử và gán giá trị cho 5 phần tử đầu tiên

int a[20] = { 1, 2, 3, 4, 5 };

mảng 2 chiều giống 1 ma trận hoạt động tương đương mảng 1 chiều

1. Chuỗi phải import thư viện string.h

* Chuỗi là một mảng các kí tự
* Cuối mỗi chuỗi luôn có một kí tự rỗng nên ví dụ khi khai báo str[20] thì thực chất cũng chỉ có 19 kí tự được viết vào chuỗi đó. Định dạng chuỗi là %s .
* Khi nhập hàm scanf với chuỗi thì thay vì là &str thì chỉ cần là str.
* Khi in một kí tự trong chuỗi thì nó cũng giống như in một phần tử trong mảng nhưng với %c.

1. Đếm số kì tự trong chuỗi

* Strlen(str): là độ dài của chuỗi, hàm này sẽ in ra một số nguyên.

1. Kiểm tra 2 xâu khác nhau

* Strcmp(xau1,xau2): kiểm 2 sâu giống hay khác nhau.

1. Hàm

* Hàm được hiểu đơn giản là 1 đoạn code, khi bạn gọi hàm thì thực ra là bạn đang gọi tới đoạn code này (bạn cũng có thể hiểu hàm là viết tắt của 1 đoạn code)
* Tới đây chắc bạn cũng đã hiểu cách tạo và sử dụng hàm, cú pháp để khai báo hàm sẽ trông giống như sau:
* void tên\_hàm() {
* /\*
* Khối lệnh bên trong thân hàm
* Khối lệnh này sẽ được thực thi khi bạn gọi hàm
* \*/
* }
* Để sử dụng hàm bạn chỉ đơn giản là gọi tới tên\_hàm();

strcmp(s1,s2): giong khi giong ca hoa ca thuong

        //strismp(s1,s2): giong nhau ca khi khac hoa va thuong.