##### 1. Con trỏ là gì?

– **Con trỏ** là một biến bình thường nhưng có thể trỏ đi mọi nơi trong bộ nhớ. Và giá trị của nó là địa chỉ của một biến khác (địa chỉ trực tiếp của một ô nhớ trong bộ nhớ).

– Về **bản chất con trỏ cũng như một biến bình thường**: nó cũng có tên biến, giá trị của biến, địa chỉ của biến. Nhưng có điểm khác là:

* Những biến bình thường thì nó chỉ nằm cố định trong 1 ô nhớ, còn biến con trỏ thì nó còn có thể trỏ đến các ô nhớ khác nhau.
* <Kiểu dữ liệu> khi khai báo cho con trỏ không phải là kiểu dữ liệu của nó, mà là kiểu dữ liệu của vùng nhớ mà nó đang trỏ đến.

##### 2. Tại sao lại cần có con trỏ?

– Vì biến con trỏ có thể trỏ đi lung tung trong bộ nhớ nên việc sử dụng bộ nhớ sẽ linh hoạt hơn.

– Áp dụng cho mảng động. Có nghĩa là khi chúng ta sử dụng mảng tĩnh với số lượng phần tử của mảng là cố định, chẳng hạn như mảng có 100 phần tử, thì dù chúng ta chỉ sử dụng 5 – 10 phần tử để thao tác tính toán thôi, thì bộ nhớ cũng sẽ cấp phát 100 ô nhớ, do đó mà sẽ gây ra lãng phí bộ nhớ không đáng có. Còn khi chúng ta sử dụng mảng động dùng con trỏ thì chúng ta sử dụng 5 thì bộ nhớ cấp phát 5 ô nhớ, sử dụng 10 thì bộ nhớ cấp phát 10 ô nhớ.

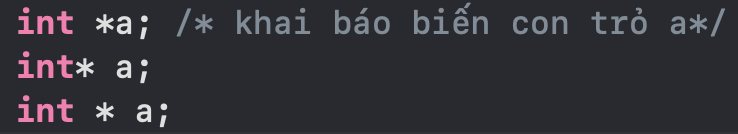
##### 3. Cách khai báo con trỏ :

– Cách khai báo :

***<Kiểu dữ liệu> \*<Tên của con trỏ>***

– Trong đó:

* **<Kiểu dữ liệu>** : Bao gồm các kiểu dữ liệu có sẵn (int, float, double, char, void) và kiểu dữ liệu do người dùng tự định nghĩa (Book, HocSinh, PhanSo,… )
* Dấu **\*** : biểu thị đây là biến con trỏ.
* **<Tên của con trỏ>** : Tuân theo quy tắc đặt tên biến trong lập trình.

Có 3 cách khai báo biến con trỏ:

##### 4. Cách sử dụng con trỏ cơ bản :

* Các biến con trỏ:

<kiểu dữ liệu> \* <tên biến>

Khi khai báo 1 biến là con trỏ thì **dấu \*** để sát tên biến.

* Toán tử con trỏ:

Có 2 toán tử được dùng với con trỏ : \* và &.

– Dùng **toán tử \*** để lấy giá trị của một biến con trỏ ( toán tử \* cũng thể hiện cho một biến là con trỏ, nên các bạn khi sử dụng cần phân biệt rõ ràng là khi nào \*p là con trỏ p, khi nào \*p là đang muốn lấy giá trị của con trỏ p ).

– Dùng **toán tử &** để lấy địa chỉ của một biến.

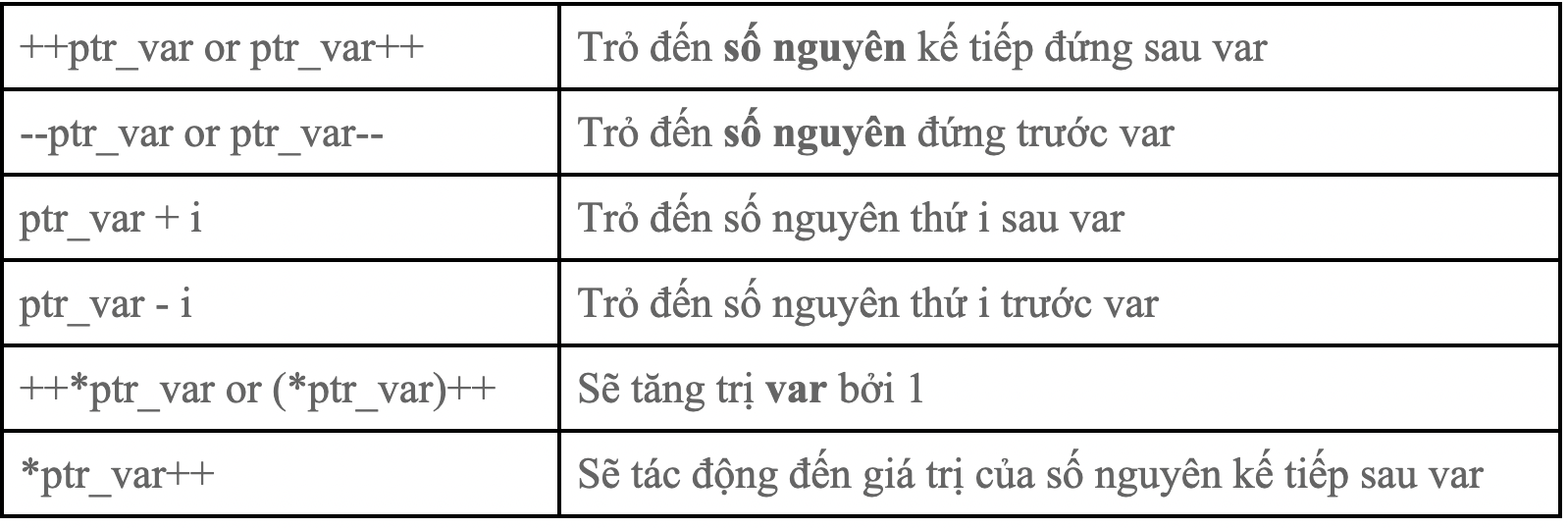
2 toán tử \* , & và toán tử lấy giá trị âm ( - ) có độ ưu tiên cao nhất trong tất cả các toán tử toán học.

* Gán giá trị cho con trỏ:

Các con trỏ cùng kiểu có thể gán cho nhau.

* Phép toán số học con trỏ:

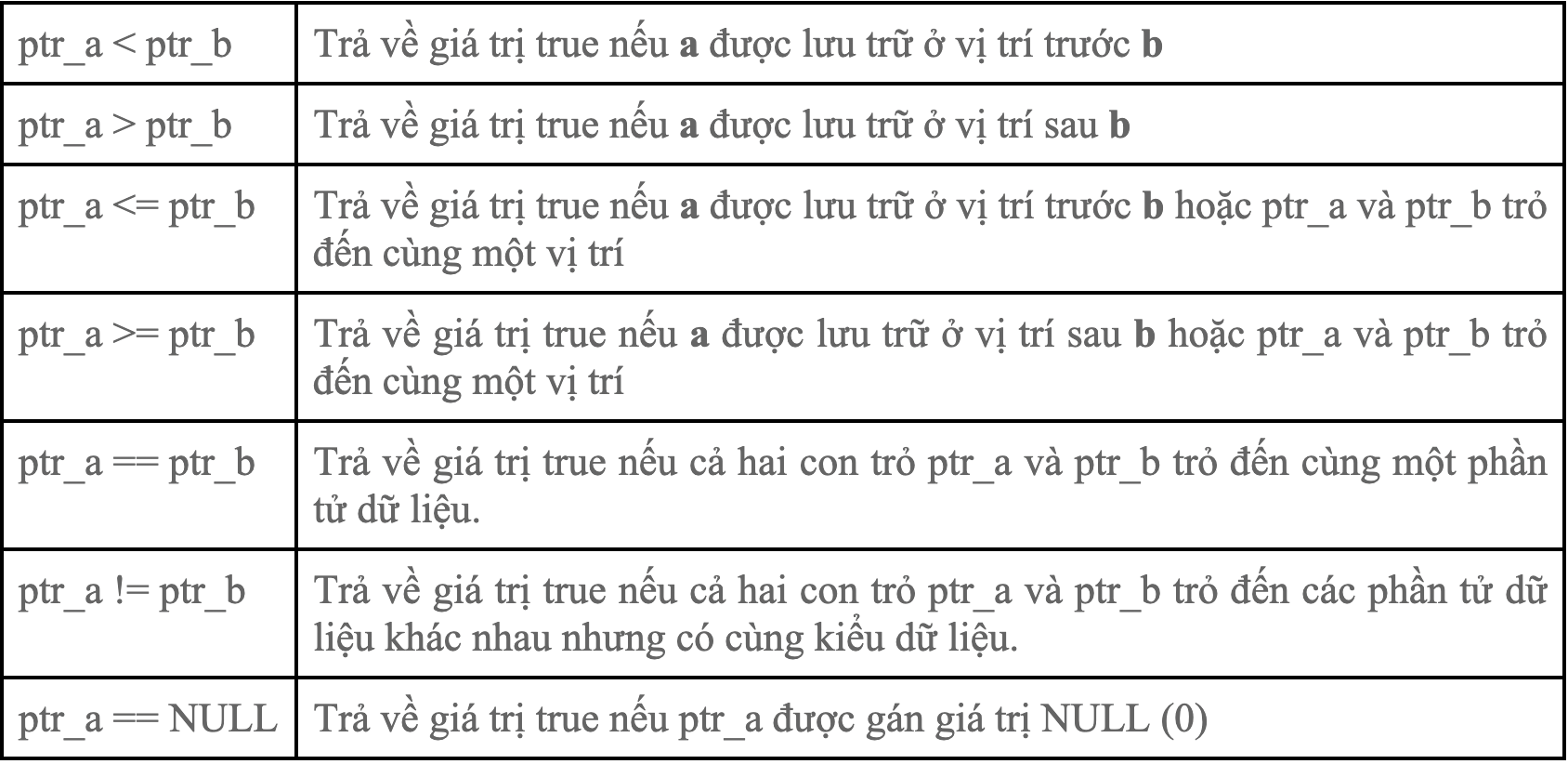
Các con trỏ chỉ có thể thực hiện các toán tử là cộng và trừ và các kiểu như float và double không dùng được với con trỏ.



Ngoài các toán tử tăng giảm giá trị, các số nguyên và ký tự cũng có thể được cộng và trừ với con trỏ. Tuy nhiên với các ký tự, những con trỏ sẽ tăng hoặc giảm giá trị tuỳ thuộc vào dộ dài kiểu dữ liệu được xác định.

* So sánh con trỏ:

Hai con trỏ có thể được so sánh trong một biểu thức nếu hai con trỏ này trỏ đến các biến có cùng kiểu dữ liệu.



Tương tự nếu ptr\_begin và ptr\_end trỏ đến các phần tử của cùng 1 mảng thì 2 con trỏ sẽ trả về số bytes cách biệt giữa 2 vị trí chúng trỏ đến.

##### 5. Con trỏ với mảng :

* Như đã biết con trỏ là 1 biến đc dùng để lưu trữ địa chỉ của dữ liệu và mảng cũng là 1 loại dữ liệu đó.
* Bằng cách sử dụng con trỏ với mảng, ta có thể truy cập vào địa chỉ của mảng và thao tác, xử lý trực tiếp với mảng này, xử lý gián tiếp mảng trong hàm qua đó nâng cao hiệu quả bộ nhớ và giảm thời gian thực thi chương trình.

Có 2 loại con trỏ mảng đó là:

1. Con trỏ và mảng 1 chiều

2. Con trỏ và mảng 2 chiều

**1.Con trỏ và mảng 1 chiều:**

Tên của một mảng nghĩa là con trỏ trỏ đến phần tử đầu tiên của mảng đó và có thể được biểu diễn theo 2 cách:

* Sử dụng ký hiệu & trước một phần tử mảng VD : &ary[i]
* Sử dụng một biểu thức trong đó chỉ số được cộng vào tên của mảng VD : ( ary+i )

Khi gán 1 giá trị cho phần tử mảng như ary[i], vế trái lệnh gán có thể được viết là ary[i] hoặc \*(ary + i) => một giá trị có thể được gán trực tiếp đến 1 phần tử mảng hoặc gán đến địa chỉ của phần tử mảng.

Trong những trường hợp cần thiết phải gán 1 địa chỉ đến 1 định danh