TRƯỜNG ĐẠI HỌC THỦY LỢI
Khoa Công nghệ thông tin
Bộ môn Tin học và KTTT

NHẬP MÔN LẬP TRÌNH

Giảng viên: TS.GVC Nguyễn Quỳnh Diệp
Email: diepnq@tlu.edu.vn
Điện thoại: 0904345673

Chương 4: Mảng, con trỏ và xâu ký tự

4.1. Mảng

- Khái niệm
- Khai báo và sử dụng
- Các thao tác thường gặp

4.2. Con trỏ

- Khái niệm và cách khai báo
- Toán tử địa chỉ (&), toán tử nội dung (*)
- Phép toán trên con trỏ
- Con trỏ và mảng

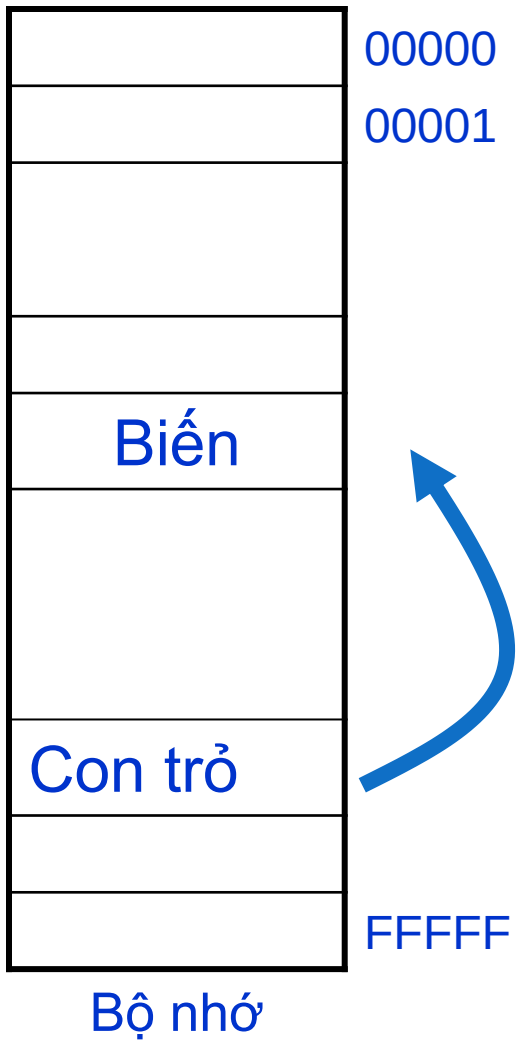
4.3. Xâu ký tự

- Khái niệm, khai báo và sử dụng
- Các hàm xử lý ký tự và xâu ký tự
- Mảng xâu ký tự

4.2. CON TRỎ

Giới thiệu

- Là một khái niệm "*mạnh*" trong C
  - Cho phép tính toán trên con trỏ
  - Sử dụng con trỏ hàm
- Cho phép truy nhập gián tiếp tới một đối tượng có địa chỉ (*biến, hàm*)
  - Truy nhập trực tiếp→ thông qua tên



Địa chỉ

- Bộ nhớ gồm dãy các ô nhớ
  - Mỗi ô nhớ là 1 byte
  - Mỗi ô nhớ có một địa chỉ riêng
- Các biến trong chương trình được lưu tại vùng nhớ nào đó trong bộ nhớ
- Khi khai báo biến, tùy thuộc vào kiểu, biến sẽ được cấp một số ô nhớ liên tục nhau
  - VD: Biến int được cấp 4 bytes, float được cấp 4 bytes,..
  - Địa chỉ của biến, là địa chỉ của byte đầu tiên trong số các byte được cấp
  - Khi gán giá trị cho biến, nội dung các byte cung cấp cho biến sẽ thay đổi

Ví dụ về địa chỉ

Địa chỉ của một biến là địa chỉ byte nhớ đầu tiên được cung cấp cho biến để lưu trữ dữ liệu

Ví dụ:

```c
#include<stdio.h>
int main(){
    int n = 1000, m = 2000, p = 3000;
    printf("Dia chi cua n = %d: %x",n, &n);
    printf("\nDia chi cua m = %d: %x",m, &m);
    printf("\nDia chi cua p = %d: %x",p, &p);
    return 0;
}
```

```
Dia chi cua n = 1000: 61fe9c
Dia chi cua m = 2000: 61fe98
Dia chi cua p = 3000: 61fe94
--------------------------------
```

Ví dụ về địa chỉ

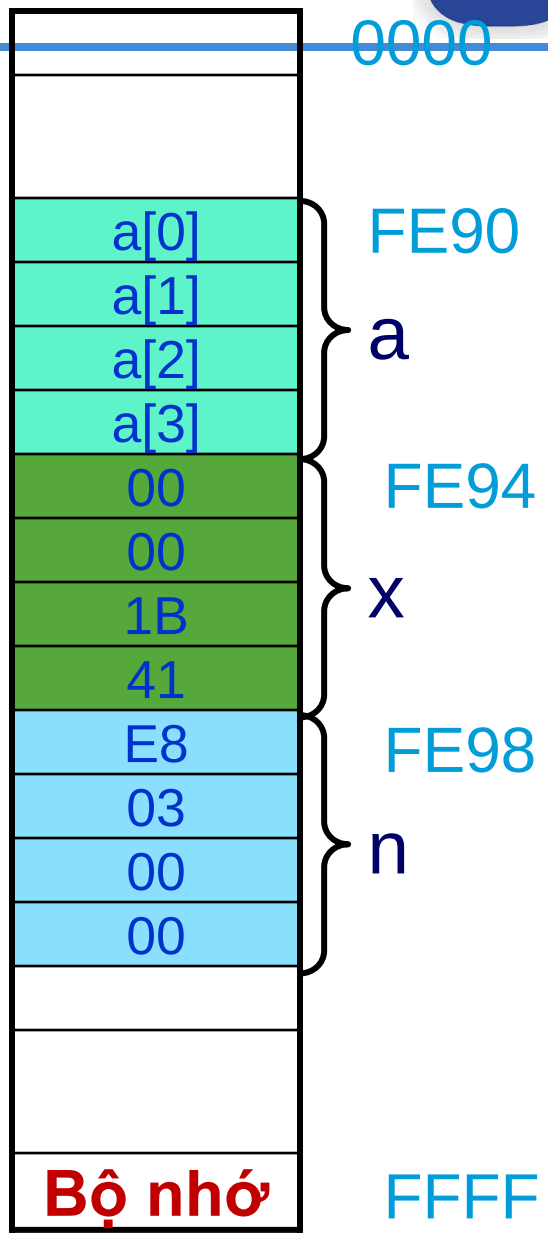
Địa chỉ của một biến là địa chỉ byte nhớ đầu tiên được cung cấp cho biến để lưu trữ dữ liệu

```
int n;
float x;
char a[4];
n = 1000; //03E8(16)
x = 9.6875; //411B0000(16) theo IEEE752/85, 32 bit
for(i=0; i<4; i++)
        a[i] = 4*i+1;
```

Ví dụ về địa chỉ

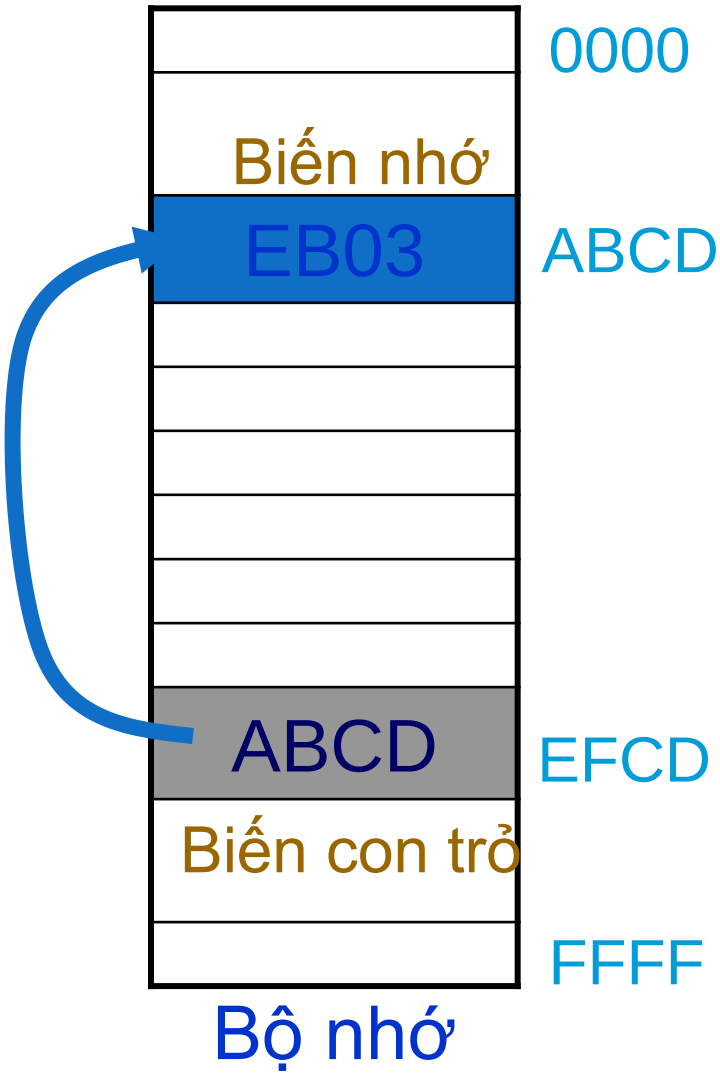
```c
#include<stdio.h>
int main(){
    int n = 1000;
    float x = 9.6875;
    printf("Dia chi cua n = %d: %x",n, &n);
    printf("\nDia chi cua x = %.4f: %x",x, &x);
    char a[4];
    int i;
    for (i = 0;i<4;i++)
    {
        a[i] = 4*i+1;
        printf("\nDia chi cua a[%d] = %d: %x",i, a[i], &a[i]);
    }
    return 0;
}
```

```
Dia chi cua n = 1000: 61fe98
Dia chi cua x = 9.6875: 61fe94
Dia chi cua a[0] = 1: 61fe90
Dia chi cua a[1] = 5: 61fe91
Dia chi cua a[2] = 9: 61fe92
Dia chi cua a[3] = 13: 61fe93
--------------------------------
```



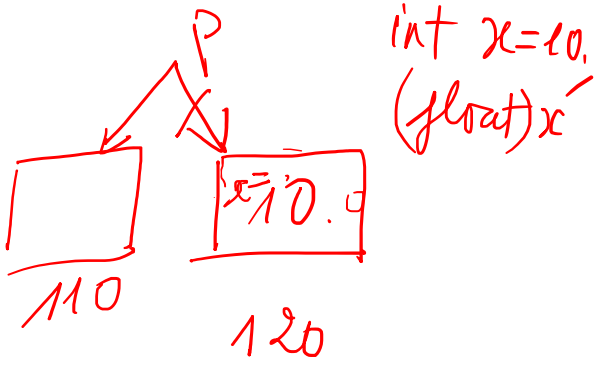
Con trỏ

- Con trỏ là một biến mà giá trị của nó là địa chỉ của một vùng nhớ
  - Vùng nhớ này có thể dùng để chứa các biến có kiểu cơ bản (nguyên, thực, ký tự,…) hay có cấu trúc (mảng, bản ghi,..)
- Con trỏ dùng "trỏ tới" một biến nhớ
  - Có thể trỏ tới một hàm
  - Có thể trỏ tới con trỏ khác



Khai báo biến con trỏ

Kiểu *Tên;

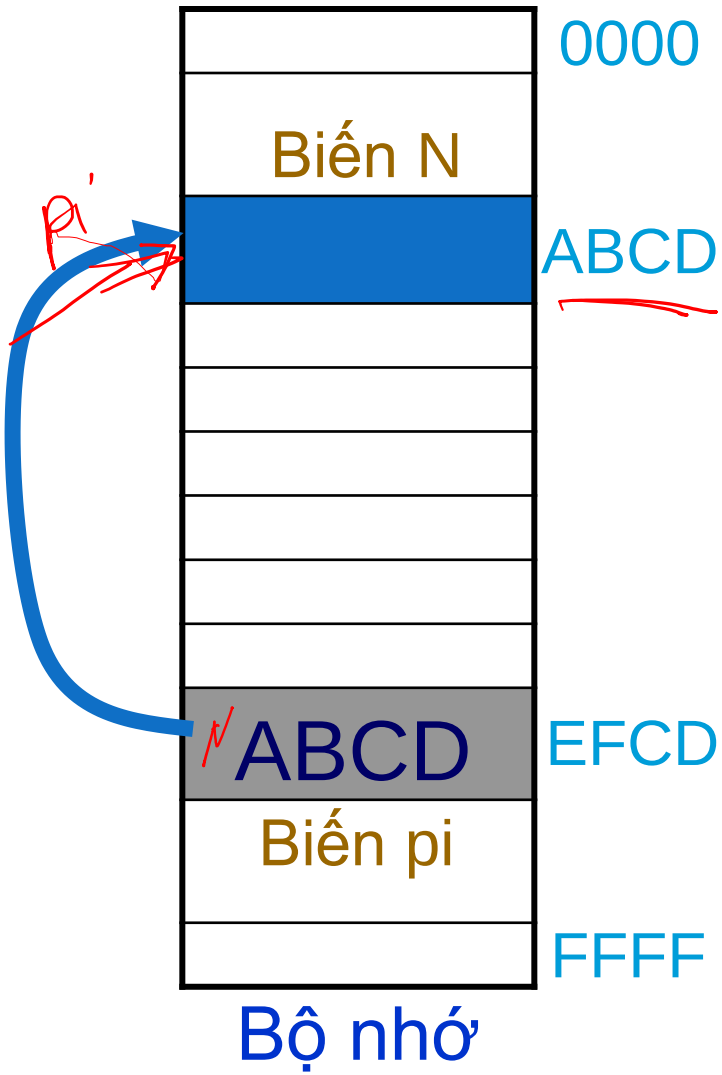


- Tên: Tên của một biến con trỏ
- Kiểu: Kiểu của biến mà con trỏ "Tên" trỏ tới
  - ➢Giá trị của con trỏ có thể thay đổi được
    - Trỏ tới các biến khác nhau, có cùng kiểu
  - ➢Kiểu biến mà con trỏ trỏ tới không thay đổi được
    - Muốn thay đổi phải thực hiện "ép kiểu"
- Ví dụ:
  - int *pi; //Con trỏ, trỏ tới một biến kiểu nguyên
  - char *pc; //Con trỏ, trỏ tới một biến kiểu ký tự

Toán tử địa chỉ (&)

- Ký hiệu: &
- Là toán tử một ngôi, trả về địa chỉ của biến
  - Địa chỉ biến có thể được gán cho một con trỏ, trỏ tới đối tượng cùng kiểu
- Ví dụ

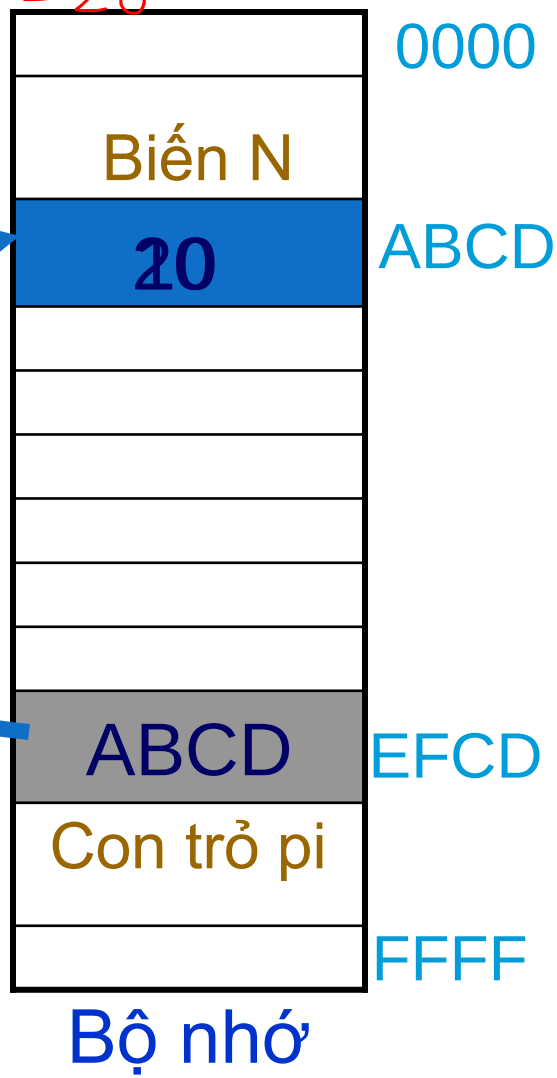
```
short int N; // &N→ ABCD
short int *pi;
pi = &N; // pi←ABCD
```



Toán tử nội dung (*)

- Ký hiệu: *
- Là toán tử một ngôi, trả về giá trị (nội dung) của vùng nhớ mà con trỏ đang trỏ tới
- Ví dụ
  - short int N;
  - short int * pi;
  - pi = &N;
  - N = 10;//Vùng nhớ mà pi trỏ tới mang giá trị 10; Vậy *pi=10
  - *pi = 20;// Vùng nhớ pi trỏ tới được gán giá trị 20; Vậy N= 20

N= 10; ⇔ *pi = 10;



Gán giá trị cho con trỏ

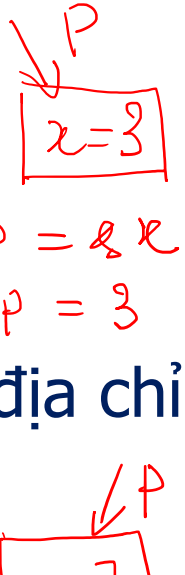
- Con trỏ được gán địa chỉ của một biến
  - Biến cùng kiểu với kiểu mà con trỏ trỏ tới
    Nếu không, cần phải ép kiểu
- Con trỏ được gán giá trị của con trỏ khác
  - Hai con trỏ sẽ trỏ tới cùng một biến (do cùng địa chỉ)
  - Hai con trỏ nên cùng kiểu trỏ đến
    Nếu không, phải ép kiểu
- Con trỏ được gán giá trị NULL

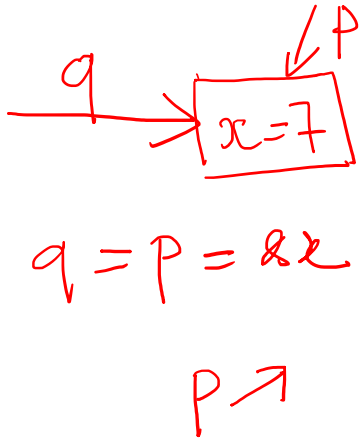
  Ví dụ: int *p;

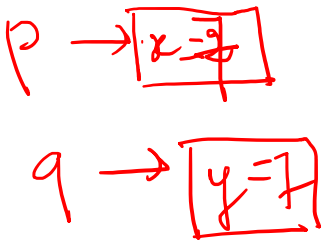
  p = 0;
- Gán nội dung vùng nhớ mà 2 con trỏ trỏ tới.

  Ví dụ: int *p1, *p2;

  *p1 = *p2;

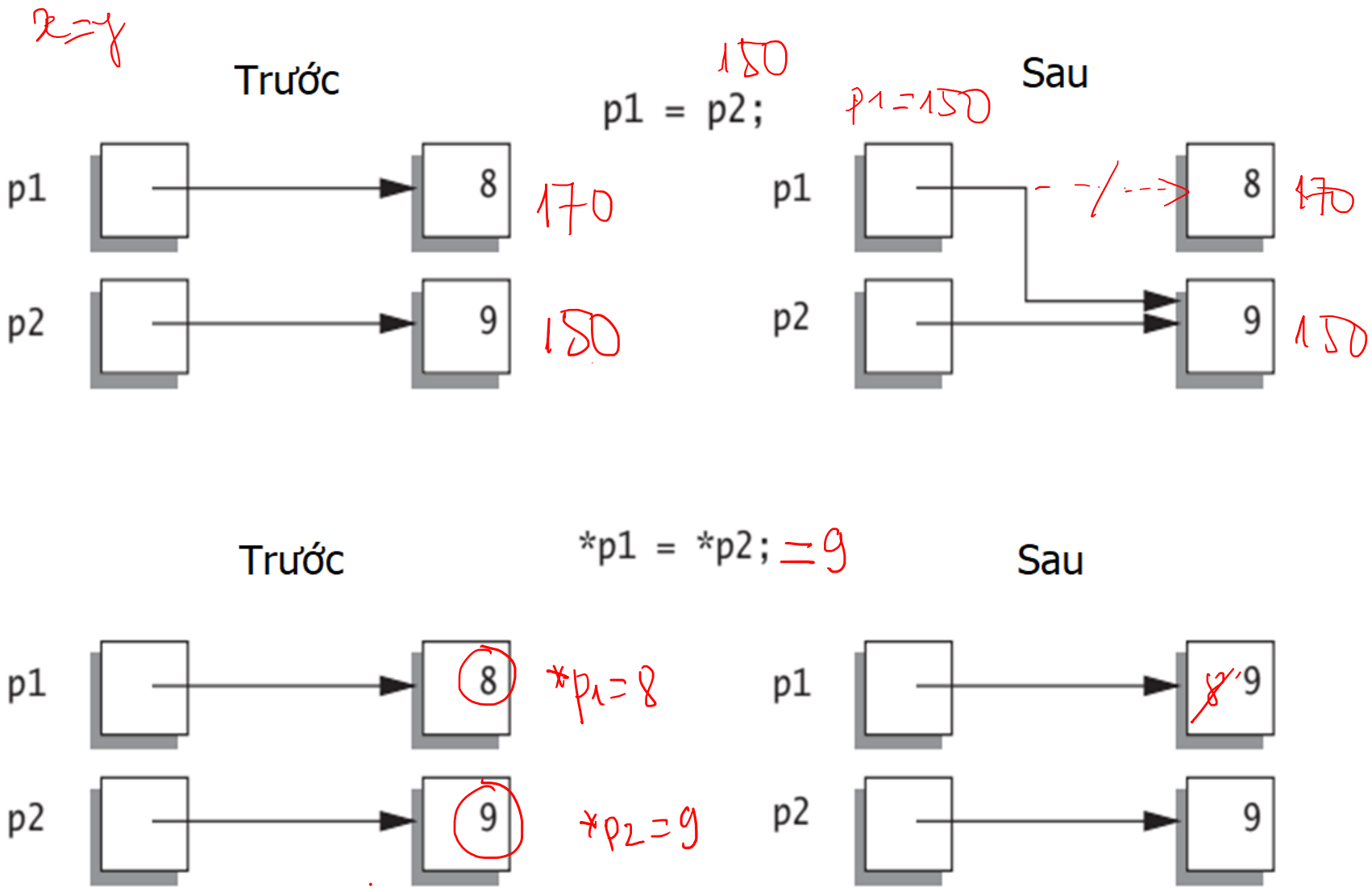






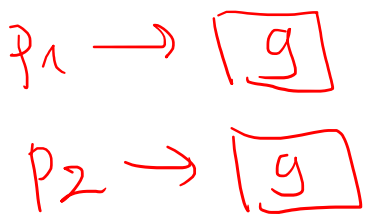
*p = *q

Gán giá trị cho con trỏ



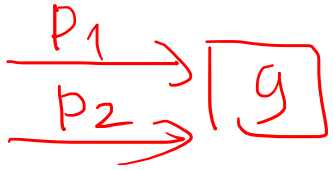
Ví dụ

```
#include <stdio.h>
int main(){
    int N = 8, M = 9;
    int *p1 = &N;
    int *p2 = &M;
    *p1 = *p2;
    printf("%d %d",*p1,*p2);
}
```



9 9

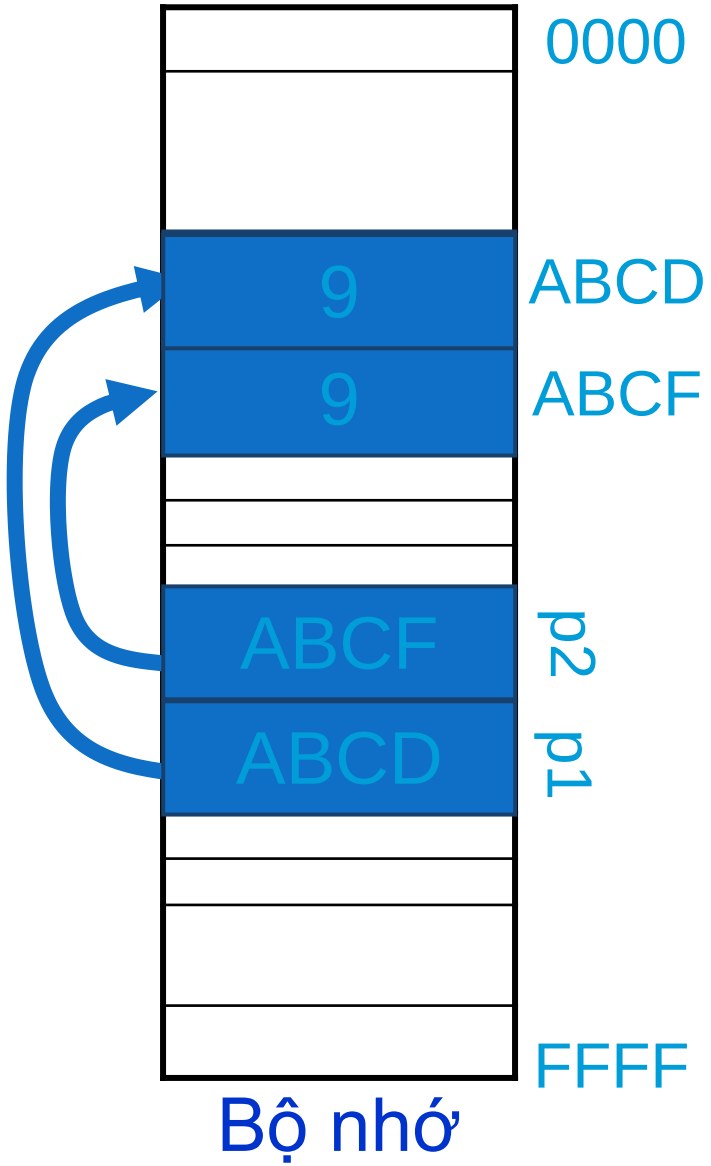
```
#include <stdio.h>
int main(){
    int N = 8, M = 9;
    int *p1 = &N;
    int *p2 = &M;
    p1 = p2;
    printf("%d %d",*p1,*p2);
}
```



9 9

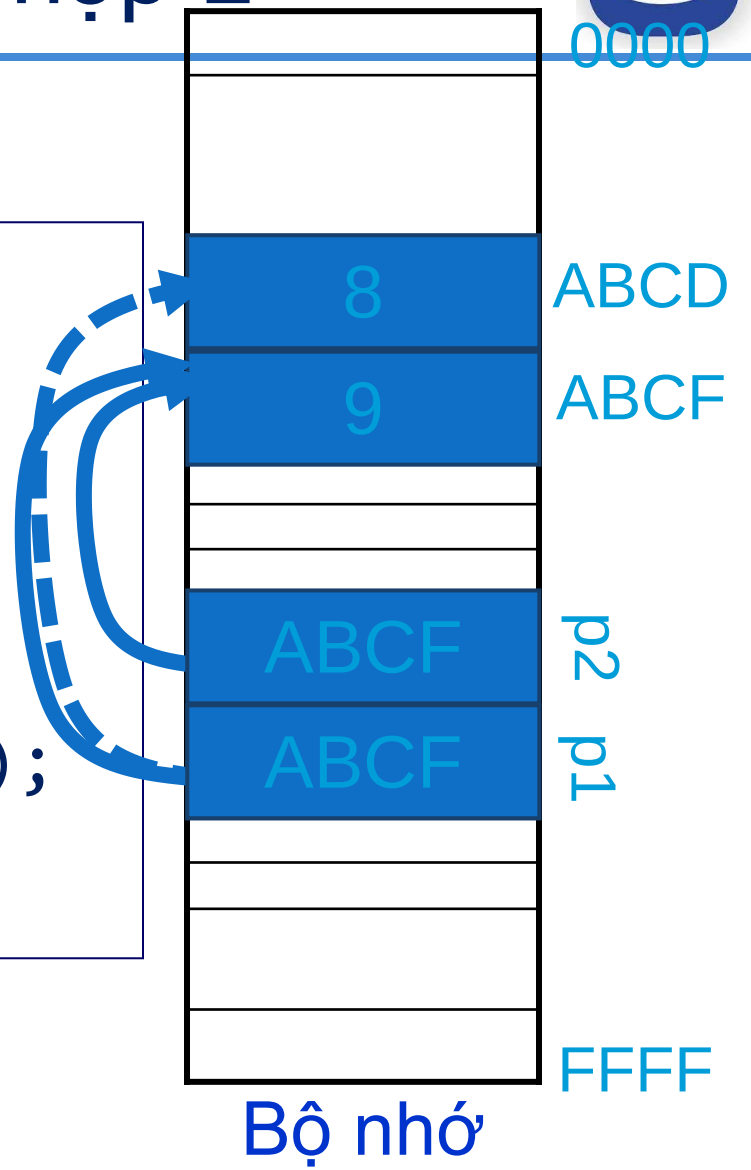
Ví dụ→Trường hợp 1

```
#include <stdio.h>
int main(){
    int N = 8, M = 9;
    int *p1 = &N;
    int *p2 = &M;
    *p1 = *p2;
    printf("%d %d",*p1,*p2);
}
```



Ví dụ→Trường hợp 2

```
#include <stdio.h>
int main(){
    int N = 8, M = 9;
    int *p1 = &N;
    int *p2 = &M;
    p1 = p2;
    printf("%d %d",*p1,*p2);
}
```



Các phép toán trên con trỏ

- Cộng con trỏ với một số nguyên
  - Kết quả: Con trỏ cùng kiểu
- Trừ con trỏ với một số nguyên
  - Kết quả: Con trỏ cùng kiểu
- Trừ 2 con trỏ cùng kiểu cho nhau
  - Kết quả: Một số nguyên
  - Khoảng cách giữa 2 con trỏ được đo bằng số phần tử thuộc kiểu dữ liệu mà con trỏ trỏ tới

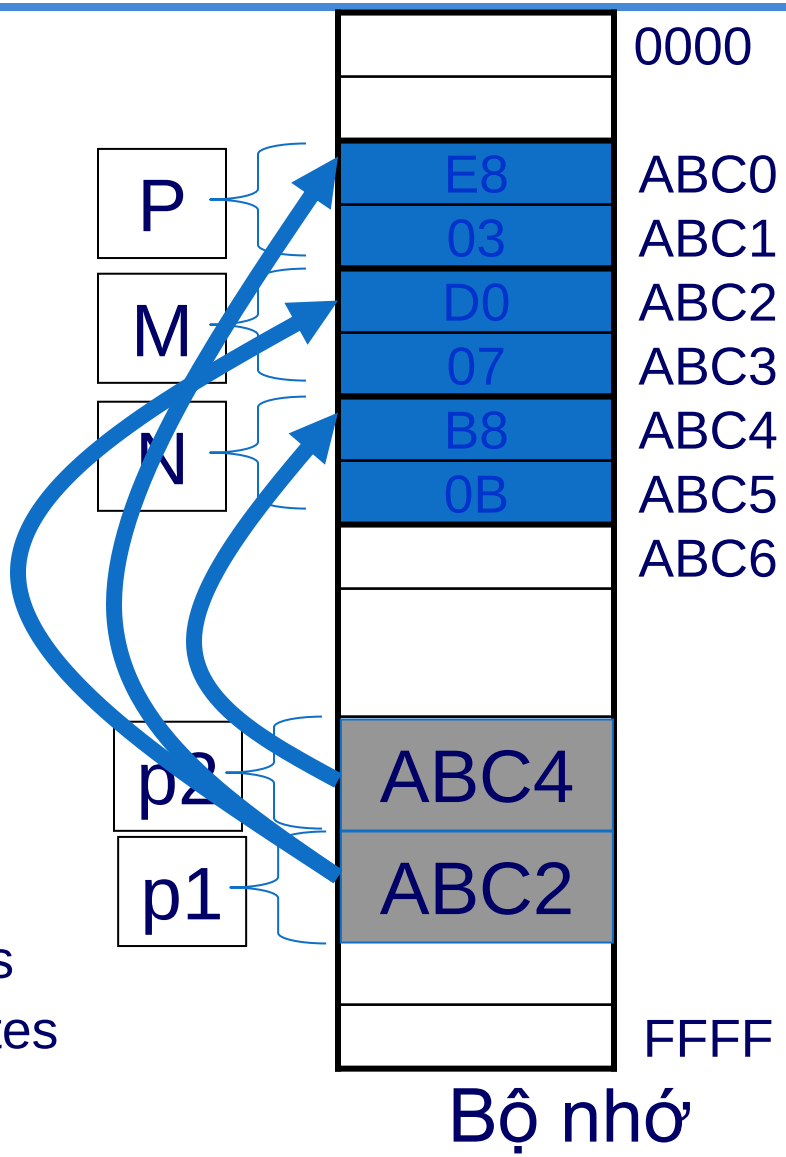
Các phép toán trên con trỏ→Ví dụ

- int N=1000, M=2000, P=3000;
- int * p1 = &P, *p2 = &N;

p1 - p2 → -2

* (p2-1) → 2000

* ++ p1 → 2000

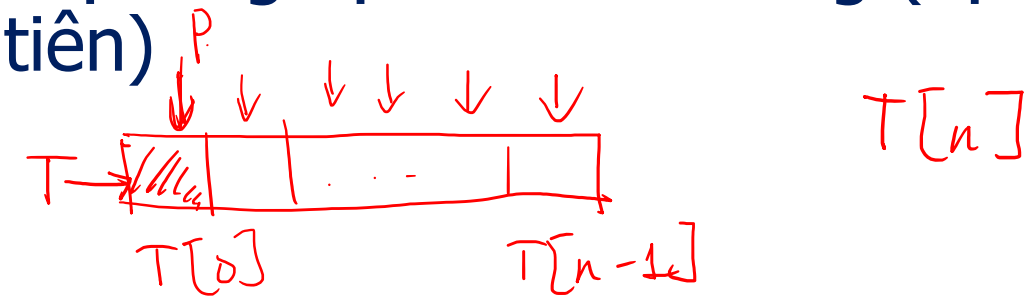


Ghi chú:

- Kiểu int, các phần tử cách nhau 4 bytes
- Kiểu float, các phần tử cách nhau 4 bytes

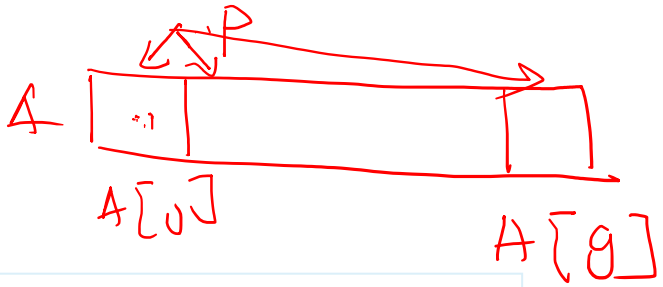
Mối quan hệ giữa con trỏ và mảng 1 chiều

- Nếu T là tên một mảng ⇒ T là một con trỏ hằng chứa địa chỉ của phần tử đầu tiên của mảng T (&T[0])
  - Không tồn tại phép tính trên tên mảng, hoặc gán giá trị cho tên mảng (Ví dụ không có T=…; T++)
- Có thể sử dụng một con trỏ để duyệt mảng nếu nó được gán giá trị bằng địa chỉ của mảng (địa chỉ của phần tử đầu tiên)



Ví dụ

```
int A[10];
int *p = A;  // int *p = &A[0]
```




```
for(i = 0; i < 10; i ++)
    printf("%d", *(p + i));
```

```
for(i = 0; i < 10; i ++)
    printf("%d", p[i]);
```

(*p): nội dung con trỏ p

p: địa chỉ mà ptrỏ tới

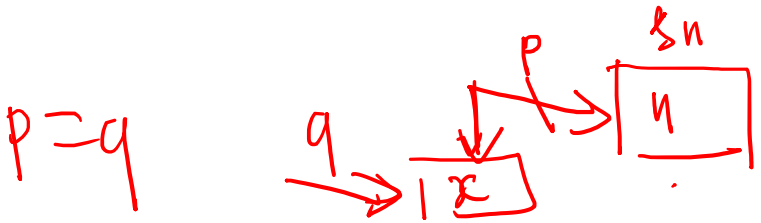
```
for(i = 0; i < 10; i ++)
    printf("%d", *(p++));
```

Con trỏ void

void * Tên_con_trỏ

- Là một con trỏ đặc biệt: con trỏ tới dữ liệu không định kiểu.
- Có thể nhận giá trị là địa chỉ của một biến có kiểu dữ liệu bất kỳ
- Ví dụ:

```
void * p, *q;
int n;  float x;
p = &n;  q= &x; \\←Các câu lệnh hợp lệ
```



Câu hỏi 1

```
#include<stdio.h>
int main(){
    int a = 3,*p;
    p = &a;
    printf("%d\n",a**p*a + *p);
    return 0;
}
```

30

Câu hỏi 2

```
#include<stdio.h>
int main(){
    int a[2][2][2] = {10, 2, 3, 4, 5, 6, 7, 8};
    int *p, *q;
    p = &a[1][1][1];
    q = (int *)a;
    printf("%d, %d\n", *p, *(q+4));
    return 0;
}
```

8, 5